5-2022

Privacy-Preserving Blockchain-Based Registration Scheme for AV Parking System

Alexander Haastrup
The University of Southern Mississippi

Privacy-Preserving Blockchain-Based Registration Scheme for AV Parking System

by

Alexander Haastrup

A Thesis
Submitted to the Honors College of
The University of Southern Mississippi
in Partial Fulfillment
of Honors Requirements

May 2022

Approved by:

Ahmed Sherif, Ph.D., Thesis Advisor,
School of Computing Sciences and Computer Engineering

Sarah Lee, Ph.D., Director,
School of Computing Sciences and Computer Engineering

Ellen Weinauer, Ph.D., Dean
Honors College

ABSTRACT

Autonomous Vehicles (AV) are a prime example of how innovation and automation are at the forefront of growing technology trends. The concern of parking systems is becoming apparent as research into ways to increase the efficiency and cost-effectiveness of AV continues. To ward against various internet attackers and secure users' sensitive information, an efficient AV parking system must have powerful user privacy and cyber security capabilities. In my work, I present a blockchain-based privacy registration system for AV parking systems that meets the following criteria. The proposed scheme incorporates k-Nearest Neighbor (kNN) - an efficient and lightweight algorithm - for encrypting and matching available parking slots of participating AV parking lots with the parking spaces of interest to AV users using vector matrices. Additionally, the incorporated blockchain eliminates the need for financial third parties and ensures secure payment fairness and transparency between the AV and parking lot. The proposed approach is also shown to be robust and efficient, according to our security and privacy analysis.

Keywords: Blockchain, Parking Reservation, Autonomous Vehicles (AV), k-Nearest Neighbor (kNN), Parking Cloud Server (PCS)

DEDICATION

This work is dedicated to God, my family, friends, fans, and haters.

Everyone, stay awesome :)

ACKNOWLEDGMENTS

I would like to extend my heartfelt acknowledgments to my advisor Dr. Ahmed Sherif for his relentless mentorship and guidance during the duration of this thesis project and to Mr. Muhammad Hataba, who eagerly shared his wealth of knowledge to aid my thesis journey.

Big thanks to the Honors College at the University of Southern Mississippi for providing me with this opportunity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

AV	Autonomous Vehicles
kNN	k-Nearest Neighbor
AVP	Automated Valet Parking
TA	Trusted Authority
PL	Parking Lots
PCS	Parking Cloud Server
BC	Blockchain

CHAPTER I: INTRODUCTION

Vehicles with autonomous capability open a world of possibilities and benefits. Sensors, machine learning systems, actuators, and multiple algorithms and processors are all included in AVs, allowing them to perform a variety of tasks like scene identification, path planning, and autonomous navigation while maintaining connection with the driver/user (Huang et al., 2018). Most of today's parking systems were designed with non-autonomous vehicles in mind. However, efficient parking systems in AVs offer a lot of benefits.

To start with, these parking systems would provide a safer mode of parking for users, ensuring both their safety and the safety of nearby pedestrians. AV parking lots have also proven to be an effective method of reducing traffic congestion. A potential parking system seeks to alleviate driver/user stress by utilizing Automated Valet Parking (AVP), which can be used to provide on-demand parking services (Kato et al., 2015). Furthermore, there is a guaranteed possibility of increased innovative infrastructure utilization. Because AVs are part of multimodal and sharing mobility systems, millions of square kilometers that are currently used for parking spaces will be freed up and diverted into areas for other valuable operations (Duarte & Ratti, 2018).

While parking schemes are being developed, it is also critical to address privacy schemes that address cybersecurity and user privacy. This is due to the increased risk of cyber exposure with hacks that could expose users' information as vehicles become more autonomous and internet enabled. Sensitive information such as a user's last known location, address, travel schedule, transportation patterns, and debit/credit card details could be obtained and used fraudulently or maliciously to take advantage of the user. This thesis addresses the question of an efficient way to conduct secure AV parking without disclosing the user's information by offering a privacy-preserving blockchain-based registration scheme.

Various blockchain-based schemes ensure service payment fairness and enable users to securely search through encrypted data. Our proposed scheme provides an efficient way to provide a registration scheme for parking in autonomous vehicles by implementing secure aspects of blockchain technology and utilizing various network and security aspects from a few previously proposed schemes. It should be noted that this thesis presents a registration scheme rather than a payment scheme. This encompasses enrolling AVs in available parking slots while maintaining privacy and concealing their data from both involved and unwanted parties. As a result, our system will consist of multiple vehicles looking for available parking spaces in various parking lots. Therefore, there is a strong need to incorporate a lightweight searching technique into our scheme. This is where the k-Nearest Neighbor (kNN) encryption algorithm comes into play.

In this thesis, our scheme combines the kNN encryption technique with an existing blockchain technique to provide a secure and privacy-aware registration scheme for an AV parking system. By implementing a single key for single encryption of the data vectors and allowing servers to perform the task of matching the encrypted vectors, our registration scheme eliminates a large count of calculation and communication overheads. We provide concrete details about this scheme and explain how the proposed scheme meets security requirements. We also put the design into action to demonstrate its feasibility as well as its performance strength.

CHAPTER II: RELATED WORKS

The use of blockchains across various industries has garnered broad interest. It provides a means for operations to be conducted in a decentralized manner and without any trusted intermediary and central authority. The extensive use of cryptography, a key feature of blockchain networks, lends authority to all network interactions. These concepts are integrated by smart contracts, which are self-executing scripts that exist on the blockchain and allow for appropriate, distributed, heavily automated workflows (Christidis & Devetsikiotis, 2016). As shown in Fig. 1, A blockchain is a synchronized and distributed data structure consisting of various sets of nodes or interconnected blocks that are replicated and shared among members who are connected through a network medium (Deng & Gao, 2020).

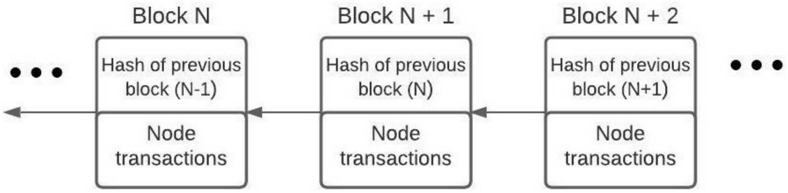


Fig. 1. Illustration of the interconnected nodes of a blockchain

Blockchain has been incorporated into countless schemes to provide security and payment fairness. In one paper, Yan et al, (2020) proposed a bitcoin-based encryption scheme that solved the problem of cloud server trust and put forward a verifiable fuzzy keyword search that uses a fuzzy keyword search retrieval scheme. It also incorporated an Ethereum smart contract to verify search results and achieve transparent service-payment fairness between the cloud server and the user. The scheme, however, suffered from comparable high computation overhead. Our scheme extracts certain aspects from the system implemented by Yan et al. (2020) and augments it to produce an efficient solution. In another paper, a scheme was proposed to provide a privacy-preserving authentication scheme using the lightweight kNN encryption technique which was shown to help verify users' authenticity; it included a design focused on non-intrusiveness, low latency, and cost-efficiency (Hataba et al., 2021). However, there was no blockchain involvement for secure storage of the authentication result. For our thesis, we develop our solution by retaining the blockchain technique used to make the payment verification from the scheme proposed by Yan et al. (2020), but change the cryptographic technique used to search over the encrypted data. To make this efficient performance refinement, we then introduce the kNN technique from the scheme proposed by Hataba et al. (2021)

Additionally, Sherif et al. (2017) proposed a scheme for ridesharing organization and incorporated a similarity measurement technique over encrypted data, representing an area where rides would be shared in the form of vectors and cells. While their scheme was not centered around creating a blockchain-based registration scheme as is ours, we also incorporate a similar themed technique, illustrated in Fig. 2, from their system to ensure the vectors are matched in our model by representing the available parking spaces from the parking lots and the interested parking slots from the AVs in the form of binary vectors

Finally, our paper takes advantage of smart contracts, which help guarantee payment fairness and transparency without the involvement of third parties. Trusted payment execution is conducted by the triggered smart contract codes on the blockchain, ensuring neither party will be cheated out of the transaction once initiated (Liu & Liu, 2019). The blockchain will arrange the smart contract between the AVs and the parking lots.

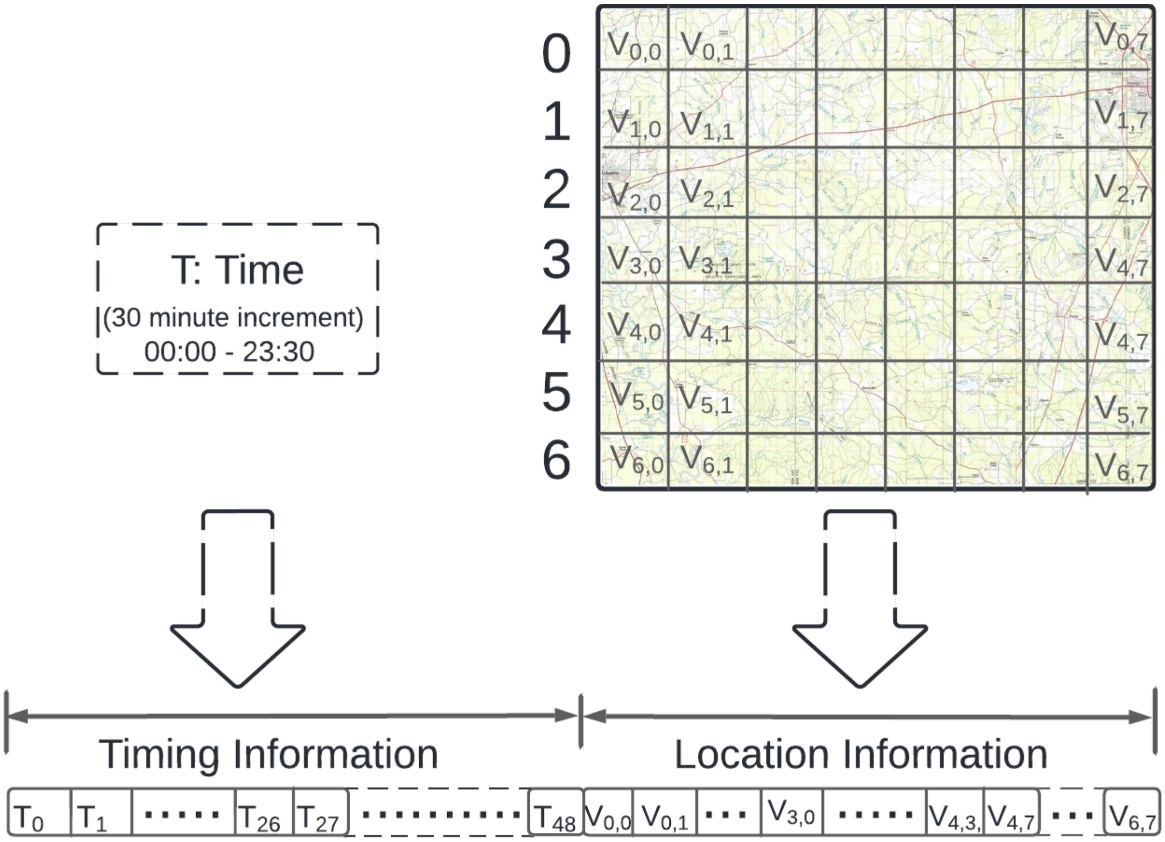

T: Time
(30 minute increment)
00:00 - 23:30
0
1
2
3
4
5
6
$V_{0,0}$ $V_{0,1}$ $V_{0,7}$
$V_{1,0}$ $V_{1,1}$ $V_{1,7}$
$V_{2,0}$ $V_{2,1}$ $V_{2,7}$
$V_{3,0}$ $V_{3,1}$ $V_{4,7}$
$V_{4,0}$ $V_{4,1}$ $V_{4,7}$
$V_{5,0}$ $V_{5,1}$ $V_{5,7}$
$V_{6,0}$ $V_{6,1}$ $V_{6,7}$
Timing Information
Location Information
T_0 T_1 ..... T_{26} T_{27} .... T_{48} $V_{0,0}$ $V_{0,1}$ ... $V_{3,0}$ ..... $V_{4,3,}$ $V_{4,7}$ ... $V_{6,7}$

Fig. 2. Timing and location vector representation

CHAPTER III: SYSTEM MODEL AND DESIGN GOALS

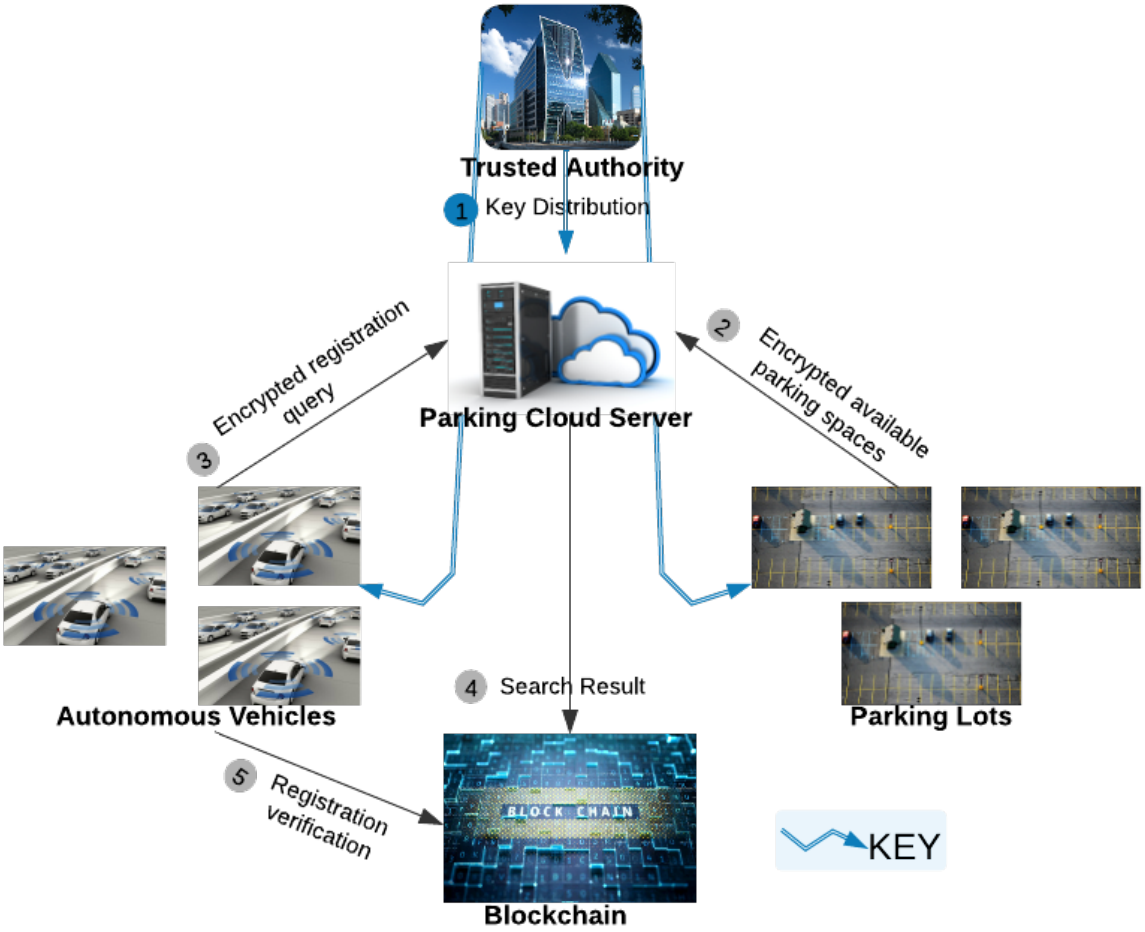


Fig. 3. The Considered Network Model

A. Network Model

As illustrated in Fig. 3, the considered network model consists of five main entities: offline Trusted Authority (TA), Autonomous Vehicles (AV), Parking Lots (PL), Parking Cloud Server (PCS), and the cryptocurrency Blockchain (BC). The TA should disseminate a unique secret key to the PCS, each participating AV, and each participating PL. Each PL should then send information on available parking slots (P) to the PCS, which will be encrypted E(P), and all this information should then be stored at the PCS.

Any AV that wishes to park and wants to search for available parking slots should send a request (M) which will also be encrypted to the PCS. Finally, an encrypted registration query E(M) should be created with the vector data of the AV who wants to park their vehicle and send this encrypted query to PCS. The cloud server should then use the encrypted registration query to look up the vector data. This will be done by computing a similarity score between the AV query and PLs' available parking data and then submitting the result to the blockchain. The AV should then verify the registration result from the blockchain. Upon being added to the blockchain, the AV should be required to deposit a certain amount of currency for payment to guarantee the correctness of the transaction. If the AV finds that the search results from their query are perfect for their request, the deposited charge will be withdrawn from their account by the blockchain, and then the AV can go ahead to register for the parking. Otherwise, the blockchain will refund the AV's deposit, and there will be no parking activity. Through this scheme, there should be no information shared between the AV and the parking lot, and there is no means for any sensitive information to be revealed.

B. Threat Model

There is the possibility of external and internal attacks on our system entities such as parking cloud server, parking lots, and AVs. Sensitive and classified information, such as encrypted available parking slots and registration queries data, are of interest to our possible attackers. While they conduct their attacks against our proposed scheme, they do so sincerely and without any intentions to cause any disturbance to the operations of our proposed scheme. Therefore, we classify our plausible attackers as "honest but curious". In our scheme, a particular attack model we consider is that of the Known Ciphertext.

Our scheme involves the searchable encrypted parking slots and registration queries information being provided by the parking lots and autonomous vehicles, respectively. Hence, in this model, the attacker is limited to the possibility of gaining access to only these sorts of information.

C. Design Goals

Based on the above threat, the design goals, which our proposed blockchain-based privacy-preserving registration scheme should attain, are as follows:

1) Registration Query Search Over Encrypted Parking Slots Vector Data from Several Parking Lots. Our scheme should be able to utilize the encrypted registration queries from the AVs to search over the encrypted vector data on available slots sent by the various participating PLs.

2) Scalability and Efficiency. The scheme should be capable of performing search operations over a large number of encrypted vectors and responding to the autonomous vehicle's queries in a timely fashion. Furthermore, communication and computational overhead should be minimized efficiently from the size of the registration query.

3) Slots and Registration Query Confidentiality. The parking cloud server should not attain any sensitive information about the stored available parking slots or the sent registration queries/requests.

4) Registration Query Unlinkability. The parking cloud server should be unable to ascertain whether two registration queries contain vector data that are identical or not.

5) Fairness. A smart contract is introduced to achieve payment fairness and transparency. The blockchain should ensure that the participating PLs do not take money from the participating AV user until the parking slot matching and registration/reservation are successfully made.

CHAPTER IV: PROPOSED SCHEME

This section delves into the proposed scheme to explain the components in detail. Before introducing the specific scheme, we provide a brief overview of the kNN technique and the vector matching model.

A. kNN Overview

kNN is a symmetric encryption technique that provides the ability to carry out similarity search or measurement over encrypted data. A kNN query searches a database for the k points that are closest to a given query point q (Wong et al., 2009). kNN brags of a lightweight distinguishing feature which allows for less computational and communication overhead compared to other encryption techniques that also perform similarity over encrypted data. Sherif et al. (2018) found in research, that the incorporation of the kNN technique in data aggregation schemes for user privacy led to lower encryption/decryption times by evaluating its experimental performance with those of existing homomorphic encryption-based schemes.

For kNN encryption and decryption, binary vectors are incorporated as splitting vectors to split the data vectors of each participating secondary user into two random vectors before a conditional operation is conducted to set the split vectors to two random numbers such that their summation is equal to one of the vectors. To end the processes, the vector pairs would then be encrypted into an index (Sherif et al., 2017). An in-depth view of the incorporation of the kNN technique in our scheme is given in following subsections of the proposed scheme below.

The main notations found in this paper are explained in Table 1 below.

MAIN NOTATIONS

Notation	Description
TASK	Trusted authority secret key
V	Secret binary vector
$\{P_1, P_2, Q_1, \ldots, Q_8\} \in R^{k\times k}$	Server secret matrices
CSSK	Cloud server secret key
PL_x	Parking Lot x
$PLSK_x$	PL_x's secret key
$\{A_x, B_x, C_x, D_x\} \in R^{Z\times Z}$	Random matrices for PL_x
AV_y	Autonomous Vehicle y
AVSKy	AV_y's secret key
$\{I_y, J_y, K_y, L_y\} \in R^{Z\times Z}$	Random matrices for AV_y

TABLE I: Main Notations

B. Vector Matching Model Overview

As shown in Fig. 2, the data being encrypted will be in form of vector (U) and the vector will be created related to the city's geographical location. According to the location, we will divide the city into different cell and each cell will be represented by a bit: "0" or "1". The timing vector will also be created which will be divided in increments of 30-minutes for each party (48 in total) and be concatenated to the location vector. Firstly, the PL_x will have its cell vector with the same number of cells and according to the physical location of the PL_x, it will then select its actual location indicating its available parking slots and available times. The choice made by the PL_x will be reflected by a "1". After generating this vector, the PL_x will then encrypt its vector by using its secret key before sending it to the cloud server as the "Encrypted Available Parking

Spaces." Secondly, the parking location and time where the AV_y will need to be decided. The AV_y will choose a specific cell location indicating its interested parking space(s) and another for its interested parking time.

The choice(s) made by the AVy will be reflected by a "1" in the chosen cell location(s). The AV_y will then send the encrypted vector that represents which locations and time they are interested in parking to the cloud server as the "Encrypted Registration Query." It may hold more than one given cell and there can only be one matching. After that, the matching will be calculated to ensure that the AV_y finds and registers with a specific PL_x in the desired geographical areas which will be represented by cells.

C. Vector Initialization

For the initial system process, the following set of oracles are run in sequential order by the trusted authority:

1) Key Distribution

First, the security parameter 1^z serves as an input for the system setup algorithm, which then outputs the TA secret key (TASK), where TASK = $\{V, P_1, P_2, Q_1, \dots, Q_8\}$, z is the size of the available parking lot slots vector, and V is a random binary vector of length z, and a set of random invertible matrices $\in \mathrm{R}^{z\times z}$.

Next, the TA generates the Cloud server secret key to be (CSSK), where CSSK = R and R is an invertible random matrix $\in \mathrm{R}^{z\times z}$. The TA also generates a PL secret key (PLSK_x) for each parking lot with available slots in the system with identity Pl_{ex}, as

$\mathrm{PLSK}_x = \{V, RQ_1^{-1}A_x, RQ_2^{-1}B_x, RQ_3^{-1}A_x, RQ_4^{-1}B_x,$

$$RQ_5^{-1}C_x,\ RQ_6^{-1}D_x,\ RQ_7^{-1}C_x,\ RQ_8^{-1}D_x\} \qquad (1)$$

where $\{A_x, B_x, C_x, D_x\}$ are random matrices $\in \mathrm{R}^{z\times z}$ such that $A_x, + B_x = P_1^{-1}$, and $C_x, + D_x = P_2^{-1}$.

Finally, the TA creates an autonomous vehicle secret key (AVSK) for each autonomous vehicle in the system (AV_y), as follows

$$\mathrm{AVSK}_y = \{V, I_yQ_1, I_yQ_2, J_yQ_3, J_yQ_4, K_yQ_5, K_yQ_6, L_yQ_7, L_yQ_8\} \qquad (2)$$

where $\{I_y, J_y, K_y, L_y\}$ are random matrices $\in \mathrm{R}^{k\times k}$ such that $I_y + J_y = P_1$, and $K_y + L_y = P_2$.

2) User Registration

Before a new user can access the system, they will initially be required to register an account with the AV. The AV will take vector data of its users' interested parking choices and then encrypt each vector data using the kNN encryption technique and the AV's secret key before uploading the encrypted data to the server. The encrypted data is used in the registration process, and it includes k indices of encrypted data where k represents the total number of AV users. The AV will execute this procedure for every user.

D. Encrypting Parking Lots Available Slots Data

Each parking lot (PL_x) builds a binary row vector (U_x) to represent the number of cells inside a city. U_x will have only one bit with digit “1”, that represent the actual location of the parking lot x (PL_x). The index is created by invoking oracle *GenerateVector*().

GenerateVector (U_x, $PLSK_x$) → E_{Ux}. For a parking lot's slot vector U_x, PL_x uses the secret V to split U_x into two-column vectors U_x' and U_x'' of the same size. If the b^{th} bit of V is zero, $U_x'(b)$ and $U_x''(b)$ are set similar to $U_x(b)$, while if it is one, $U_x'(b)$ and $U_x''(b)$ are set to two random numbers such that their summation is equal to $U_x(b)$. Before submitting to the cloud server, PL_x then uses its secret key $PLSK_x$ to compute the parking slots data W_{Ux} as:

$$W_{Ux} = [\, RQ_1^{-1}A_xU'_x,\ RQ_2^{-1}B_xU'_x,\ RQ_3^{-1}A_xU'_x,\ RQ_4^{-1}B_xU'_x;\ RQ_5^{-1}C_xU''_x,\ RQ_6^{-1}D_xU''_x,\ RQ_7^{-1}C_xU''_x,\ RQ_8^{-1}D_xU''_x] \quad (3)$$

where W_{Ux} is a column vector of size $8Z\ (m * n)$, where Z is the number of city cells.

E. Encrypting Registration/Reservation Query

To ensure that a user is authenticated and registered, or the request is not from a malicious entity, each Autonomous Vehicle (AV_y) will use its corresponding secret key $AVSK_y$ delivered from the TA to encrypt a registration query for each user who wants to park his/her vehicle. This is done without revealing any of the user's information to the PL or allowing the server to access any user's sensitive information.

Before submission to the cloud server, each AV_y generates an encrypted reservation query by invoking oracle *GenerateTrapdoor*().

$GenerateTrapdoor\ (F,\ \mathrm{AVSK}_y) \rightarrow G_F$. Given the user's chosen location data vector T, AV_y uses V to split F into two random row vectors, f' and f'', which are of the same size. The splitting method is described as follows. If the b^{th} bit of V is one, $f'(b)$ and $f''(b)$ are set similar to $t(b)$, while if it is zero, $f'(b)$ and $f''(b)$ are set to two random numbers, such that their summation is equal to $f(b)$. Then, AV_y uses its secret key AVSK_y to generate the trapdoor G_F

$$G_F = [f'\,I_yQ_1,\ f'\,I_yQ_2,\ f'J_yQ_3,\ f'J_yQ_4,$$
$$f''K_yQ_5,\ f''K_yQ_6,\ f''L_yQ_7,\ f''L_yQ_8]$$

(4)

where G_F is a row vector of size $8Z$.

F. Matching Vectors Data

The parking cloud server will begin searching over encrypted available parking lot(s) data by calculating the dot product operation between the generated trapdoor from the AV with the vector indices inside its database in oracle *Match*. Here, the data will be searched through without being decrypted to further fortify security. The cloud server will incorporate the kNN encryption scheme. The cloud server will then use the search outcome to validate the autonomous vehicle's authentication query.

$Match(CSSK,\ W_{Ux},\ G_F) \rightarrow ReservationResult$. In this oracle, the cloud server should first use its secret R^{-1} to remove R from W_{Ux} to obtain $\bar{W}_{Ux}$, where

$$W_{Ux} = [\ Q_1^{-1}A_xU'_x,\ Q_2^{-1}B_xU'_x,\ Q_3^{-1}A_xU'_x,\ Q_4^{-1}B_xU'_x;$$
$$Q_5^{-1}C_xU''_x,\ Q_6^{-1}D_xU''_x,\ Q_7^{-1}C_xU''_x,\ Q_8^{-1}D_xU''_x]$$

(5)

Next, a similarity score is computed between the trapdoor G_F and the index $\bar{W}_{Ux}$ by dot product operation ($G_F \cdot \bar{W}_{Ux}$), where the dot product is represented by *(·).* The cloud server will use the trapdoor received from the AV to measure its similarity with the parking lots' indices by invoking *Match*() oracle. Then, the server sends the best-matched results to the blockchain. If there is a match with a query coming from the AV and one of the stored available parking lot indices, a positive confirmation will be observed. Incorporating our scheme, the similarity score of the indices and trapdoors can be measured by the PCS.

Proof: This can be done by computing $G_F \cdot W_{Ux}$, as follows.

$$G_F(\cdot)\bar{W}_{Ux} = f'I_yA_xU'_x + f'I_yB_xU'_x + f'J_yA_xU'_x + f'J_yB_xU_x'$$

$$+ \ f''K_yC_iU_x'' + f''K_yD_iU_x'' + f''L_xC_xU_x'' + f''L_yD_xU_x''$$

$$= f'(I_y + J_y)(.)(A_x + B_x)U_x' + f''(K_y + L_y)(.)(C_x + D_x)U_x''$$

$$= f'P_1P_1^{-1}U''_x + f''P_2P_2^{-1}U''_x$$

$$= F(\cdot)U_x$$

G. Store Results at Blockchain and Verification

Finally, the result of the matching will be sent to the blockchain for storage and verification. Ethereum, being a programmable blockchain system and a decentralized application platform, could be implemented for the contract which will be written in the blockchain in digital format. The smart contract is a collection of agreements that serve to regulate the digital assets and it consists of the responsibilities and privileges of the transaction participants. Once certain previously set conditions are satisfied, it is executed automatically and does not require any intermediary intervention. Upon receiving the match results, $G_F(\cdot)W^{-}_{Ux}$, the verification smart contract gets deployed.

For verification, the blockchain will have some holding on the AV and if the AV finds that the search results is perfect for his request, the blockchain will take the money from their account and then the AV can make the successful parking reservation. However, if there matching results are not suitable for the AV user, the blockchain will release the money holding from their account, thereby ending the transaction with neither the AV nor the parking lot losing any monetary resource. How will the blockchain be used to store and verify the result.

CHAPTER V: PRIVACY ANALYSIS

According to our design goals described in section III-C, in this section, we show that our proposed scheme satisfies all the described privacy capabilities and requirements by addressing each listed goal.

*1) Registration Query Search Over Encrypted Parking Slots Vector Data from Several Parking Lots***.** Our proposed scheme can use the encrypted registration/reservation queries sent from an AV to search over the encrypted vector data on available slots sent by the various participating parking lots. In our scheme, the AVs do not need to share their private key with participating parking lots to search over their encrypted parking slots data. The AV can use the same key given by the TA to search all the vector data from the participating parking lots.

2) Scalability and Efficiency. Our proposed scheme can perform search operations over a large number of encrypted vectors from numerous PLs and respond to the AV's queries quickly. In our scheme, this is possible because only a dot product operation of the involved vectors is computed. Furthermore, the communication and computational overhead in our scheme are comparably feasible.

3) Slots and Registration Query Confidentiality. The PCS cannot attain any information that may prove sensitive or valuable about the stored available parking slot or the sent registration queries. In our scheme, this is possible due to the data encryption from the AV and PL. This enables vector and reservation/registration query confidentiality and ensures that the data is never uploaded to the PCS in plaintext. Data privacy is further preserved because data is never decrypted on the server as the PCS searches exclusively over encrypted data.

4) Registration Query Unlinkability. The PCS cannot ascertain whether two registration queries contain data that are identical or not. In our scheme, the registration trapdoor unlinkability stays intact as different ciphertexts will be given if the same trapdoor is transmitted multiple times. Additionally, without the parking cloud server's security key, $CSSK$, the similarity scores computed by the PCS cannot be determined by eavesdroppers.

5) Fairness. The reliability of the blockchain smart contract can make sure that payment fairness and transparency are achieved. A smart contract is introduced to achieve payment fairness and transparency. Due to this implementation in our scheme, there is a guarantee that the participating PLs will not take any money from the participating AVs until the parking slot matching and registration/reservation are made.

CHAPTER VI: EXPERIMENT AND PERFORMANCE EVALUATION

This section analyzes our proposed scheme concerning the experimental setup and performance analysis.

A. Experiment Setup and Evaluation Metrics

For performance evaluation, our proposed scheme was implemented using Python and a server with an AMD Ryzen 5 3500U, Radeon Vega Mobile Gfx 2.10 GHz processor, 8.00 GB of RAM, and a 64-bit operating system running a Windows 10 Home Operating System. In our experiments, for our participating parking lots available slots and interested autonomous vehicles, we created a concatenated unit vector containing a vector sized (m x n) for the slot locations plus a vector for timing with a 30- minute increment (48 in total) for each involved party. The unit vectors were then used in the parking cloud server's vector dot product matching step.

The experiment on our proposed scheme was run with one AV user in mind and various sample sizes for the parking lots and the grid map with available parking slots. As shown in Fig. 4 and Fig. 5, the communication and computational overhead were also studied to reflect our system scalability with increased parking lots, map sizes, and potentially more AVs.

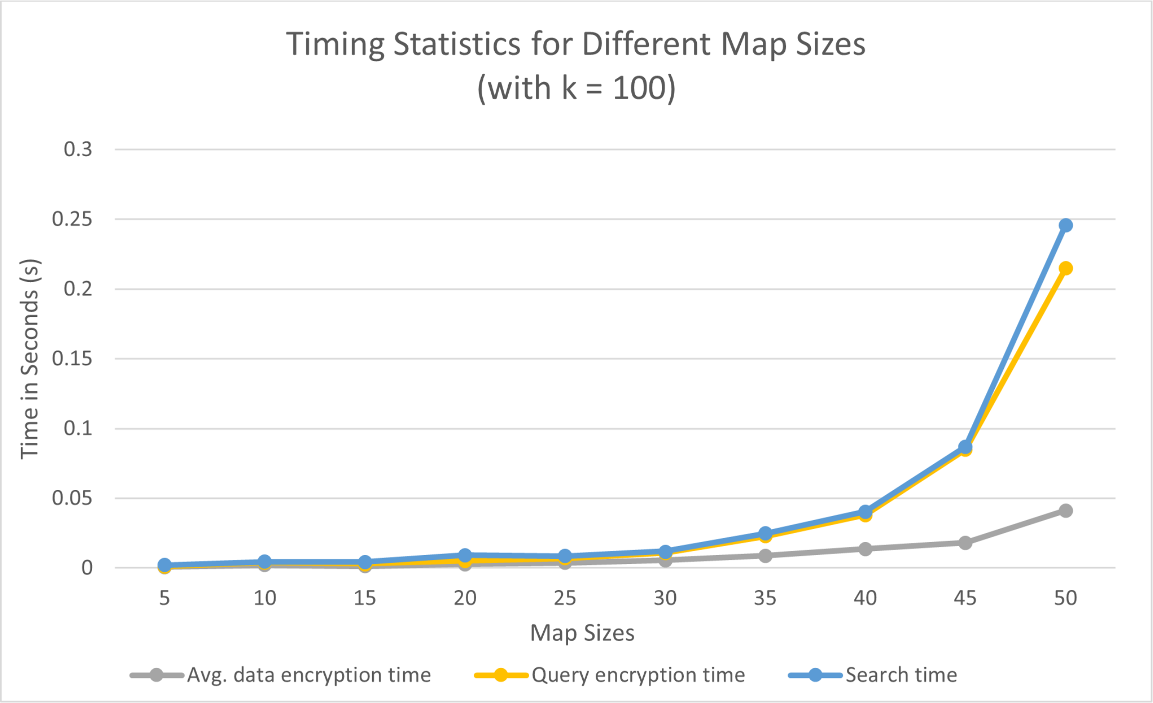


Fig. 4. Timing Statistics for Different Map Sizes

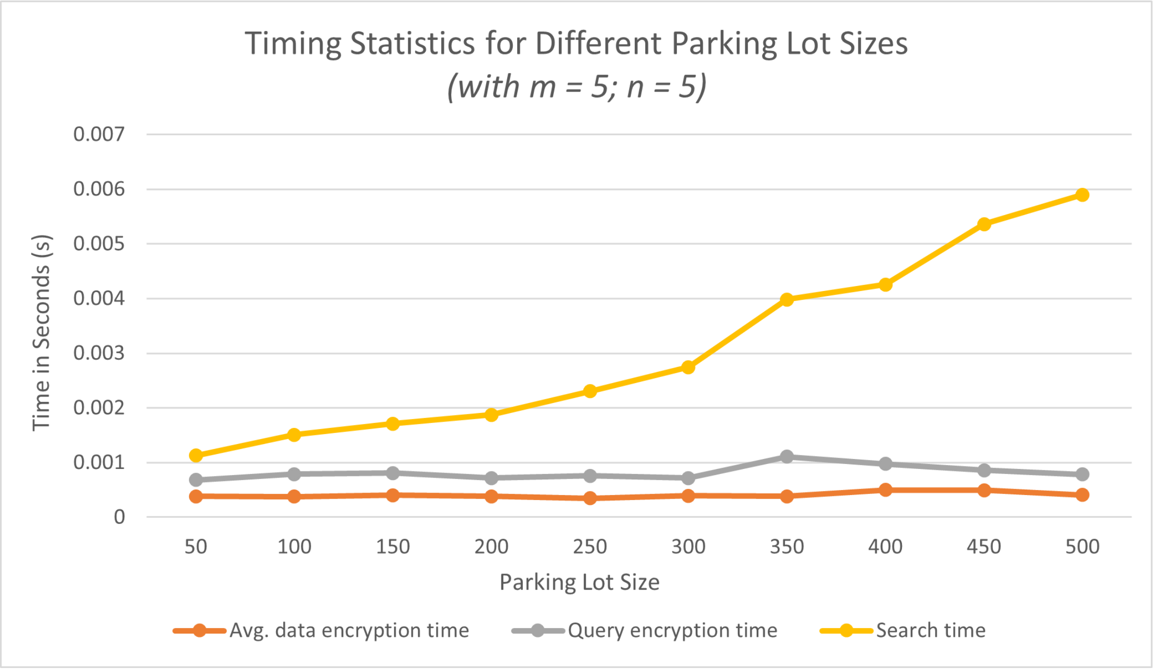


Fig. 5. Timing Statistics for Different Parking Lot Sizes

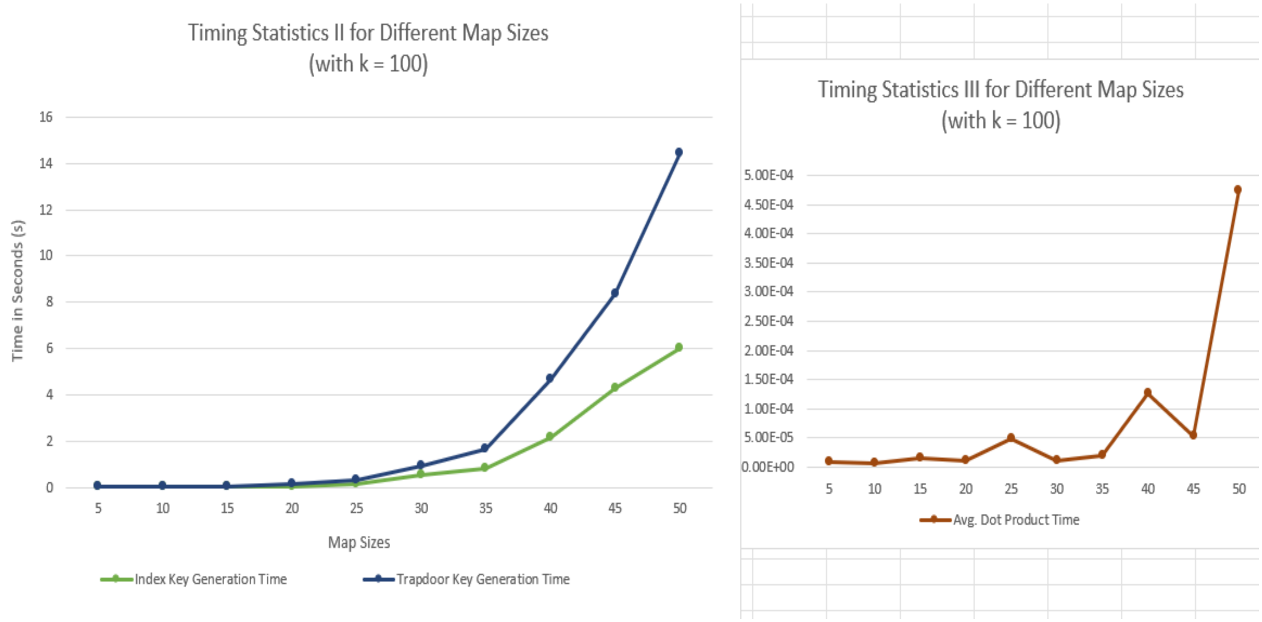


Fig. 6. Timing Statistics II & III for Different Map Sizes

B. Experiment Results

Here, the experimental analysis of our proposed scheme is given in detail concerning communication overhead and computation overhead, as well as encryption scalability and search time analysis. There are four variables involved: $T_{PLKeyGen}$, T_{AV_KeyGen}, $T_{Encrypt,}$ and $T_{DotProd}$. T_{PL_KeyGen} represents the time required for the parking lot key generation PL_x, $T_{AVKeyGen}$ represents the time required for the autonomous vehicle AV_y key generation, $T_{Encrypt}$ represents the time required for the trapdoor encryption, and $T_{DotProd}$ represents the time required for available parking slots and query trapdoor dot product computation.

1) Communication and Computational Overhead. The kNN technique brings efficiency and lightweight benefit to our scheme. This is evident in the data size communicated for each index or reservation query. For the communication and computation overhead, the durations are dependent on our map size. As illustrated in Fig. 4, our communication overload is the vector size $(m\ x\ n) + 48$, where $(m\ x\ n)$ represents the map size. Hence, we notice that as the map size increases, the time also increases, and as the map size reduces, the computation and communication time also reduces

2) Computation Overhead. Fig. 6 depicts the timing statistics for various map sizes concerning the $T_{PLKeyGen}$, $T_{Encrypt}$, and $T_{DotProd}$. From our findings, we also observe that the core of our matching process was in μsecs, and the other important processes were in $m$secs, further proving our technique's lightweight feature and its practical potential to manage upscaling in terms of data size and processing.

3) Encryption Scalability. The performance of our system was observed in gradually increasing usage scenarios. We experimented with numerous sample sizes for the parking lot size with a gradual increment of fifty. The encryption times are within a reasonable margin, as shown in Fig. 5. The figure reflected that our system's performance scale moves in an upward curving direction along with the increasing number of parking lots. Our findings also observe that with fixed map size and an increasing number of parking lots, we get quicker, scalability, and processing time statistics.

4) Search Time Analysis. From Figs. 4 - 6, we can observe the average timing needed for our parking cloud server to search over a subset of encrypted vector data with the encrypted registration query. This is done by conducting a dot product operation between the generated trapdoor and the stored vector matrix. The search process ends when the similarity scores are compared, and matching is found. As previously mentioned, we observed that the dot operations were in μsecs and the timing operations were in $m$secs, further proving the lightweight and efficiency of our scheme's technique.

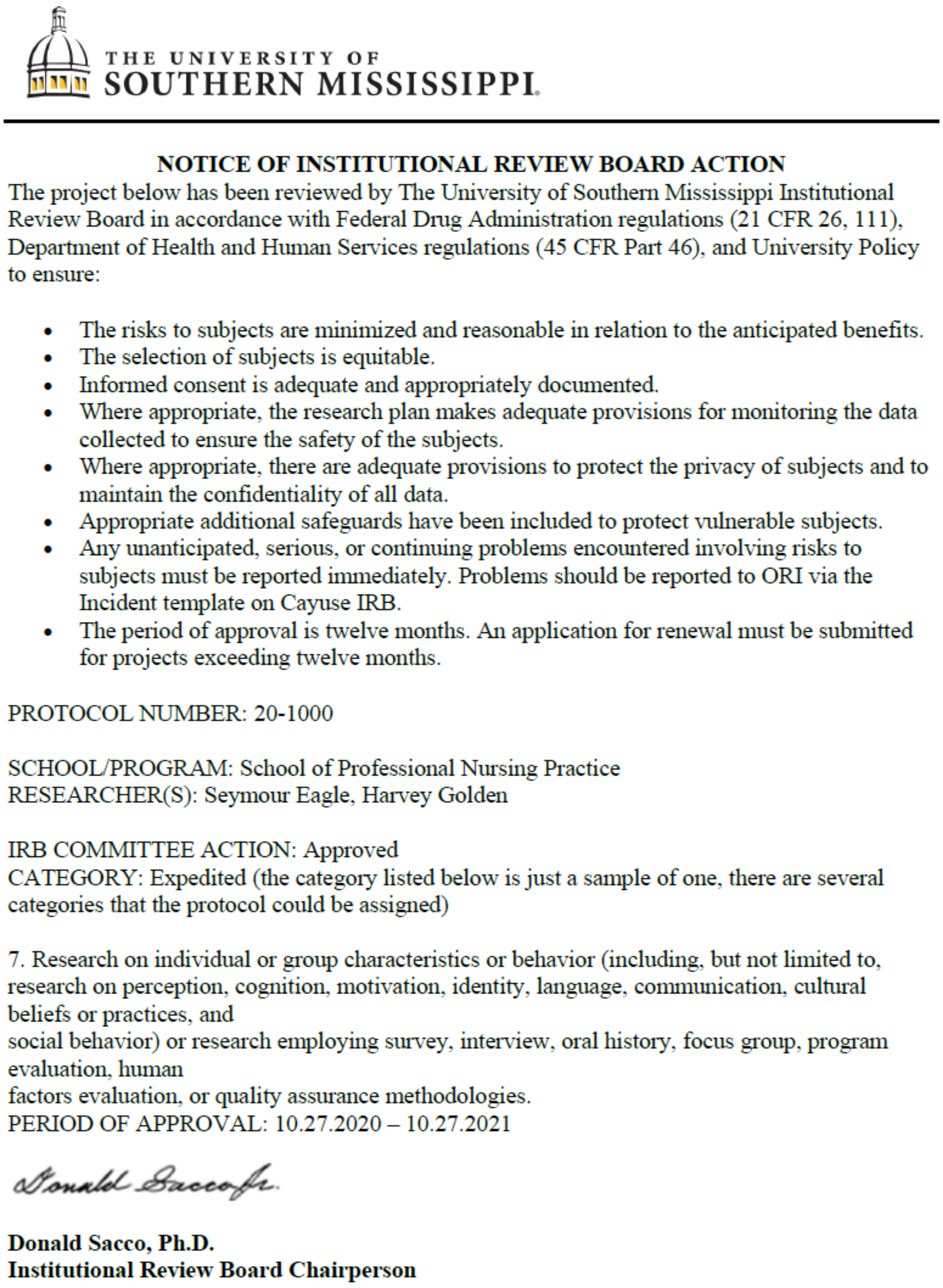
5) Fairness Analysis. The blockchain is introduced to ensure secure transaction fairness and transparency between the PLs and the AV user in our scheme. The smart contract's predefined mechanism helps to this effect. Our scheme's service-payment fairness is guaranteed further due to the blockchain's irreversible and immutable strength. As an added benefit, this reduces the potential computational burden by eliminating the need for local verification and third-party financial intervention.

In summary, our experimental results demonstrate that our proposed scheme is relatively efficient and lightweight, making it a feasible application in blockchain-based payment schemes for AV parking systems with mathematically validated privacy awareness, unlinkability, and service-payment fairness goals.

CHAPTER VII: CONCLUSION

Security and privacy awareness are crucial aspects of any emerging technology. A secure, efficient parking registration scheme for vehicles with autonomous capabilities focused on preserving privacy and security ensures no exploitable fault in this aspect during AV parking. This thesis proposes a scheme that satisfies that effect by incorporating the encryption efficiency of the kNN encryption technique -defending even against honest-but-curious attackers- and the security and transparency of blockchain for fair payment. Parking lots' slots availability and AV users' interested parking spots are also matched smoothly through vector cell representation and matrix matching. Furthermore, the results from the dataset testing indicate that our proposed scheme has feasible computational overhead and low communication cost.

APPENDIX A: IRB APPROVAL LETTER



THE UNIVERSITY OF
SOUTHERN MISSISSIPPI.

NOTICE OF INSTITUTIONAL REVIEW BOARD ACTION

The project below has been reviewed by The University of Southern Mississippi Institutional Review Board in accordance with Federal Drug Administration regulations (21 CFR 26, 111), Department of Health and Human Services regulations (45 CFR Part 46), and University Policy to ensure:

- The risks to subjects are minimized and reasonable in relation to the anticipated benefits.
- The selection of subjects is equitable.
- Informed consent is adequate and appropriately documented.
- Where appropriate, the research plan makes adequate provisions for monitoring the data collected to ensure the safety of the subjects.
- Where appropriate, there are adequate provisions to protect the privacy of subjects and to maintain the confidentiality of all data.
- Appropriate additional safeguards have been included to protect vulnerable subjects.
- Any unanticipated, serious, or continuing problems encountered involving risks to subjects must be reported immediately. Problems should be reported to ORI via the Incident template on Cayuse IRB.
- The period of approval is twelve months. An application for renewal must be submitted for projects exceeding twelve months.

PROTOCOL NUMBER: 20-1000

SCHOOL/PROGRAM: School of Professional Nursing Practice
RESEARCHER(S): Seymour Eagle, Harvey Golden

IRB COMMITTEE ACTION: Approved
CATEGORY: Expedited (the category listed below is just a sample of one, there are several categories that the protocol could be assigned)

7. Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and
social behavior) or research employing survey, interview, oral history, focus group, program evaluation, human
factors evaluation, or quality assurance methodologies.
PERIOD OF APPROVAL: 10.27.2020 – 10.27.2021

Donald Sacco Jr.

Donald Sacco, Ph.D.
Institutional Review Board Chairperson

REFERENCES

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*. https://doi.org/10.1109/ACCESS.2016.2566339

Deng, X., & Gao, T. (2020). Electronic Payment Schemes Based on Blockchain in VANETs. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.2974964

Duarte, F., & Ratti, C. (2018). The Impact of Autonomous Vehicles on Cities: A Review. *Journal of Urban Technology*, *25*(4), 3–18. https://doi.org/10.1080/10630732.2018.1493883

Hataba, M., Sherif, A., Elsersy, M., Nabil, M., Mahmoud, M., & Almotairi, K. (2021). Privacy-Preserving Biometric-based Authentication Scheme for Electric Vehicles Charging System. 86–91. https://doi.org/10.1109/MENACOMM50742.2021.9678231

Huang, C., Lu, R., Lin, X., & Shen, X. (2018). Secure Automated Valet Parking: A Privacy-Preserving Reservation Scheme for Autonomous Vehicles. *IEEE Transactions on Vehicular Technology*, *67*(11), 11169–11180. https://doi.org/10.1109/TVT.2018.2870167

Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K., & Hamada, T. (2015). An Open Approach to Autonomous Vehicles. *IEEE Micro*, *35*(6), 60–68. https://doi.org/10.1109/MM.2015.133

Liu, J., & Liu, Z. (2019). A Survey on Security Verification of Blockchain Smart Contracts. *IEEE Access*, *7*, 77894–77904. https://doi.org/10.1109/ACCESS.2019.2921624

Sherif, A., Alsharif, A., Mahmoud, M., Abdallah, M., & Song, M. (2018). Efficient Privacy-Preserving Aggregation Scheme for Data Sets. *2018 25th International Conference on Telecommunications (ICT)*, 191–195. https://doi.org/10.1109/ICT.2018.8464922

Sherif, A. B. T., Rabieh, K., Mahmoud, M., & Liang, X. (2017). Privacy-Preserving Ride Sharing Scheme for Autonomous Vehicles in Big Data Era. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2016.2569090

Sherif, A., Alsharif, A., Mahmoud, M., & Moran, J. (2017). Privacy-Preserving Autonomous Cab Service Management Scheme. *Proceedings of the 3rd Africa and Middle East Conference on Software Engineering*, 19–24. https://doi.org/10.1145/3178298.3178303

Wong, W. K., Cheung, D. W., Kao, B., & Mamoulis, N. (2009). Secure kNN computation on encrypted databases. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, 139–152. https://doi.org/10.1145/1559845.1559862

Yan, X., Yuan, X., Ye, Q., & Tang, Y. (2020). Blockchain-Based Searchable Encryption Scheme with Fair Payment. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3002264