SAPMS: a secure and anonymous parking management system for autonomous vehicles

Oladayo Olufemi Olakanmi

Department of Electrical and Electronic Engineering,
Faculty of Technology Building,
University of Ibadan,
Office 6, Ibadan, Nigeria
Email: olakanmi@mit.edu

Abstract: Recent surveys on autonomous vehicle (AV) (SAE level 5) have shown its potential in transforming road transportation system. Its advent will revolutionise road transportation, however, before this could happen vital operations in road transportation management need re-modification or redesign. One of these operations is parking system; most of the existing parking systems are targeted towards non-autonomous vehicles, where parking is not only distance-bound but parking prices are affected by other factors such as time and location. In this paper, a smart and anonymous parking management system using a novel space selection technique and anonymous authentication for space selection and reservation is proposed. The system determines the parking pattern with the lowest parking cost using the developed space selection algorithm. The performance of the system was evaluated in terms of estimated computation cost and possibility of obtaining optimal parking pattern under the dynamic pricing system. The results showed that the proposed system is capable of selecting the best parking pattern at the lowest computational cost.

Keywords: smart parking system; autonomous vehicle; transportation management; privacy; authentication.

Reference to this paper should be made as follows: Olakanmi, O.O. (2020) 'SAPMS: a secure and anonymous parking management system for autonomous vehicles', *Int. J. Information and Computer Security*, Vol. 12, No. 1, pp.20–39.

Biographical notes: Oladayo Olufemi Olakanmi received his BTech in Computer Engineering in 2001 from the Ladoke Akintola University of Technology, MSc in Computer Science and PhD in Electrical and Electronic Engineering all from the University of Ibadan, Ibadan, in 2006 and 2014 respectively. He is a Senior Lecturer in the Department of Electrical and Electronic Engineering, University of Ibadan, Nigeria. His research interests include high performance computing, security and privacy, and embedded systems.

1 Introduction

Falling costs of radar technology and advances in machine learning enabled the potential of autonomous, self-driving vehicles. This new transportation option could affect how cities are planned, requiring fewer parking places and more zones for picking up

passengers and dropping them off. It could also increase how much people travel by car because of its lower cost and convenience. Also, autonomous vehicles (AVs) have parking peculiarities such as anticipatory, time-piece and distance independent parking which contribute to its lower cost and convenience. Therefore, good parking management system is one of the prerequisites for the adoption of AVs as means of road transportation. Many parking systems with efficient performance had been proposed (Kim et al., 2010; Lu et al., 2009; Chim et al., 2011; Mahmud et al., 2013; Lochert et al., 2008), however most of them are built around non-AVs and could not accommodate some of the parking peculiarities of AVs. For example, most of these parking systems were designed around the drivers, where initiation of parking events and responses are done by the drivers. Any instance of removing the drivers from such parking system renders it non-operational. Another major operational issue that renders some of these parking systems unsuitable for AVs is the selection criteria considered in these parking systems when searching for parking space. In most of the non-AVs parking systems, walkable distance, which is the distance between the driver's point of transit and parking lot, plays a major role in the spaces selection. That is, distance of the parking lot to the driver determines the suitability of space irrespective of its current price or availability. Meanwhile, AVs can park far from the destination inasmuch they get parking spaces and at a cheaper rate. This difference in selection criteria further shows the wide gap between the non-autonomous and AVs' parking systems.

Another major issue in parking system is ability of the system to maintain the privacy of its entities at no extra cost. Some existing parking systems use public key infrastructure (PKI), and certificate based scheme to preserve privacy of their entities specifically vehicles (Lu et al., 2009, 2012), however considering the issues raised in Ellison and Schneier (2000) about 'Who is using my key' and 'who is liable for certificate compromise'. Under some digital signature laws, once the user's signing key has been certified by an approved certificate authority (CA), then user is responsible for whatever that key does even when the key is later compromised by approved CA (Ellison and Schneier, 2000). This is a weakness on any scheme built around certificates generated by third party. The big question for future research in the parking system would be to find solutions to the above mentioned constraints in order to have a safe, efficient, robust and privacy preserving parking system for AVs.

2 Related works

Searching for parking space is one of the road transportation activities that waste time and energy especially in metropolitan cities. Several works had been done to reduce some of the problem associated with transportation management. A few of them delved into parking system for all levels of non-AVs. Examples of these could be seen in Zadeh and Dela (2016). In their work, they proposed novel solution for locating free parking space by providing motorists with real-time information of the parking space and their status. In this work, drivers could monitor available space and navigate to nearest available parking spots using the mobile application. Authors in Biondi et al. (2016) also presented smart parking system that is capable of helping drivers to locate parking slot using context-aware information to help in the process. An algorithm which compares routes of different drivers to find relations among them was used. Also, a technique based on

Bluetooth low energy advertising (BLE advertising) was adopted to detect passengers and reduce the bias that would be introduced by the arrival of multiple users in the same parking area.

In vehicular technology, security and privacy is a main issue that must be dealt with during information exchange. Preserving sensitive information during communication, charging and payment had been extensively researched and different schemes had been proposed to secure and preserve information and identity of vehicle (Au et al., 2014; Boukerche et al., 2008; Cho et al., 2013). One of these schemes was described in Li et al. (2015a). In their work, the authors presented Lynx, an authenticated anonymous real-time reporting protocol, which allowed electric vehicles (EVs) to send anonymous reports using unlinkable pseudonyms. The pseudonym is capable of being verified by the utility without compromising the true identity of the EV. It also consists of payment scheme that allows the utility to issue anonymous receipts to EVs, which can be used later by the EVs to anonymously claim credits. In same vein, a secure roaming EV charging protocol which helps preserve users' privacy was proposed in Mustafa and Zhang (2014). In this protocol a verifiable pseudonym would be used by EV during charging. This protocol protects the user's identity from other suppliers as well as the user's location from its own supplier. Also, it allows the users contracted supplier to authenticate the EV and the user.

Another privacy scheme was proposed in Zhao et al. (2014) to enhance the privacy of the drivers by using anonymous credential technique and trusted platform module (TPM). This scheme is capable of determine the charging price using the battery specification to determine the charging rate. Another scheme that preserve the privacy of the vehicle was proposed in Jung et al. (2009). In their scheme, a universal re-encryption and identity based key establishment schemes were engaged. These schemes allow RSUs to issue multiple anonymous certificates to an OBU just to reduce overhead associated with mutual authentication between OBUs and RSUs. The authors in Kokalj-Filipovc and Fessant (2010) attempted to solve privacy problem associated to social interaction amongst vehicles by introducing a trusted self-organised network infrastructure for running anonymous P2P applications.

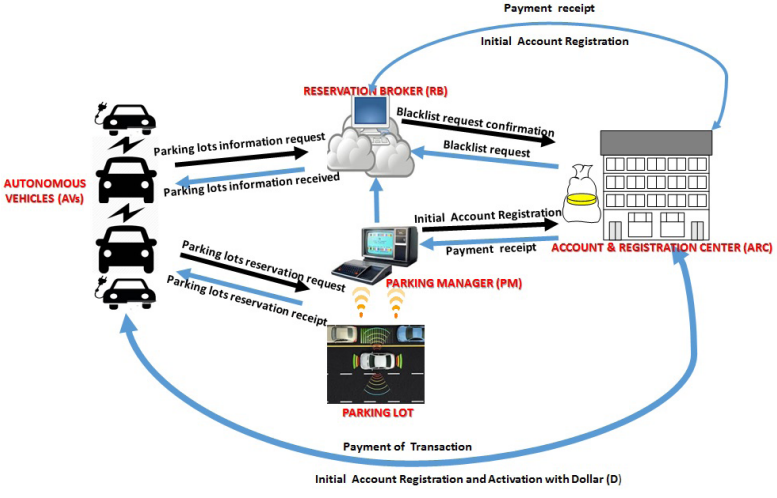
In spite of all these research efforts on vehicular technology, some important aspect of parking system concerning AV still need attention such as minimisation of parking cost, security and privacy preservation during information exchange. In this paper we propose smart and anonymous parking management system for AVs. In this parking system, all the parking lots upload their updated parking spaces' status into a centralised cloud server at every instance of a new free parking space. AV, using its verifiable pseudonym, places request to the server for vacant spaces or parking pattern that matches its need in terms of charging space and distance. It reserves parking lot based on the parking pattern generated by the space selection algorithm. The remainder of this paper is organised as follows. Section 2 discusses some of the related past works on parking system. Meanwhile, Sections 2–4 describes the system model and security requirements, design goals and objectives, and the primitives used. In Sections 5–7 design principle of the secure and anonymous parking management system is presented. In Section 8 performance evaluation and security analysis of the system is discussed while Section 9 is the conclusion.

3 System model and security requirement

3.1 System model

The proposed system consists of four essential entities as illustrated in Figure 1. These are, AV that is capable of searching for parking space, and performing single or time-piece parking; reservation broker (RB) is the central cloud server that stores different parking lots' status information and processes AVs' space search request; parking manager (PM) of each parking lot serves as interface between the parking lot and other entities, and uploads parking lot's updated spaces information into central server RB. The upload is done at every instance of a new free space or occupied space in the lot; account and registration centre (ARC) is the trusted party that generates initial authorised anonymous key for both AVs and PMs as well as performing conflict resolution in the system.

Figure 1 Model of secure and anonymous parking management system for AV (see online version for colours)



3.2 Threat model

All the threats in STRIDE threat model are considered as the major threats for our parking management system; that is spoofing, tampering with data, repudiation, information disclosure, denial of service (DoS), elevation of privilege (Microsoft, 2005).

1 *Elevation of privilege:* Adversary AV may want to maliciously use previously used or expired certificate in order to cheat during reservation and payment. Also, an adversary may try to use a stolen certificate to launch Sybil attack on an honest AV, making her paying for a reservation not made.

2 *Spoofing identity:* In this threat, we assume that adversary may want to identify AV through its past and present transactions or trace its location.

3 *Repudiation:* This involves AV denying granted reservation or unrequested reservation is forced on AV by PM.

4 *Information disclosure:* This involves RB conniving with adversary to spoof on AVs by leaking out transactions' information that could be used to obtain their real identities or locations.

5 *Denial of services:* DoS attacks deny service to valid AVs or PMs. For example, by making a PM temporarily unavailable or overwhelming RB with spurious requests.

Table 1 Symbols and their description

Symbol	*Description*
Park-ID	Parking lot identification
$\mathbb{G}_1$, $\mathbb{G}_2$	Two multiplicative group
P	Generator of elliptic curve $\mathbb{E}$
e	Bilinear function
H	One-way hash function
φ, ϕ	Private session key, public session key
AAK	Anonymous master key
t, μ	Certificate counter, key life span
ρ	Tracking list
Pkt_1	Request packet from AV to RB
Pkt_2	Reservation packet from AV to PM
$SlSg$	Short life self generated key or pseudonmy
Υ	Certificate counter
Rt	Reservation response packet from PM to AV
$Cert$	Certificate
σ_{av}	Signature of AV
σ_{pm}	Signature of PM
F_{pm}	Pseudonym of PM
F_{av}	Pseudonym of AV
SAE	Society of Automotive Engineers
γ	Parking cell identity
$CP_t(I \mapsto j)$	Cost incur by AV to perform time-piece parking from parking lot i to j
$d_{(j \mapsto i)}$	Distance from location j to i
$dc_{(j \mapsto i)}$	Cost of moving from location j to i
$Pw_t(j \leftrightarrow i)$	Cost of parking from lot j to i
$Pw_t(j \leftrightarrow j)$	Cost of parking in lot j

Table 2 Search packet format

$SlSgK_j$	2-bit C_v	12-bit Maximum distance (M_d)

3.3 Design goals and objectives

Our design goals are centred on provision of a parking management system for AVs. This system takes advantage of the peculiarities of AVs to reduce parking cost and other performance bottlenecks such as security and privacy issues associated with information exchange in parking system. Our major goals are highlighted below.

1 *Cost and efficiency:* SAPMS should be able to reduce the cost and waiting time using the proposed space selection algorithm. The algorithm optimally allows AV to select a parking pattern with the least parking cost.

2 *Anonymity and integrity:* SAPMS preserves the anonymity of the AV and integrity of the exchange message during search and reservation phases. This is to prevent an adversary from tracing the AV's present and future locations.

3 *Scalable traceability:* SAPMS through a trusted party should be capable of unveiling the real identity of dishonest AVs and PMs for blacklisting.

4 *Time-piece parking:* Since distance is no longer an issue in parking AVs, a time-piece parking pattern where an AV can switch between parking lots for minimum parking cost is proposed as an alternative to single parking pattern.

Our design objectives in order to achieve the above goals involve:

- Parking cost and waiting time optimisation for AVs through a space selection algorithm that allows timepiece and single parking. In this objective, space selection algorithm which reduces waiting time and parking cost is developed.
- Conditional anonymous authentication: An anonymous authentication scheme is developed for the parking management system. This is used to preserve the privacy of AV during transaction such as search, reservation and parking.
- Decentralised self-generated keys and certificates: A decentralised key generation is proposed for the anonymous authentication scheme such that each entity is capable of generating its subsequent keys without any computation overhead on ARC.

4 Primitive

Two multiplicative cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ are employed by all the entities to generate their self-generated keys and certificates. $\mathbb{G}_1$, $\mathbb{G}_2$ are of the same prime order q where g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. $\mathbb{G}_1$, $\mathbb{G}_2$ are asymmetric groups if there exists a bilinear map e: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, such that $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ $\forall a, b \in Z_q^*$. The isomorphism from $\mathbb{G}_1 \mapsto \mathbb{G}_2$ is denoted as $\hbar$, which is assumed to be a one-way function. Also there exist an additive group of elliptic curve $\mathbb{E}$ of generator P which is used to create session key for each entity.

The SAPMS strength depends on one way characteristic of the used hash function, discrete logarithm problem (DLP) and elliptic curve discrete logarithm problem (ECDLP) as described below.

Definition 1: (One-way hash function)

There exists a secure one-way hash function H: $X \rightarrow Y$, where $X = 0, 1^*$ and $Y = Z_p^* = \{a|0 < a < p \text{ and } \gcd(a, p) = 1\}$ satisfying the following requirements:

- For a given $y \in Y$, it is hard to find an $x \in X$ such that $H(x) = y$.
- For a given $x \in X$, it is hard to find another $x' \in X$, with $x' \neq x$, such that $H(x') = H(x)$.
- It is hard to find a pair $(x, x') \in X \times X$, with $x' \neq x$, such that $H(x') = H(x)$.

Definition 2: DLP

Let p be a prime, and $g \in Z_p$ of large order p. The function $c \rightarrow g^x mod p$ seems to be one-way. In other word, given c and g, it is difficult to find an integer x' and c' such that $c' \equiv g^{x'} mod p \equiv c$.

Definition 3: (ECDLP)

Here the ECDLP problem which the certificate generation of the scheme also depended on can be formally defined as follows: Suppose E is an elliptic curve over $\mathbb{Z} = p\mathbb{Z}$ and $P \in E(\mathbb{Z} = \mathbb{Z}_p)$, given a multiple Q of P, the ECDLP is to find $n \in \mathbb{Z}$ such that $nP = Q$.

5 A smart and anonymous parking management system for AVs

The design principle of the propose parking management system is described in this section. It consists of four phases; registration, anonymous space searching, space selection. Each of these phases are described below.

5.1 Registration phase

Every new entity (AV or PM) registers with the ARC. The basic function of the ARC is the provision of anonymous authentication parameters. The registration phase involves session pair key setup, master key setup, and shortlife-self-generated (SlSg) key and certificate generation and pseudonym generation stages:

5.1.1 Session key setup

A session pair key (φ, ϕ) is generated by each entity using elliptic curve with $\mathbb{E}(\mathbb{Z} = \mathbb{Z}_p^*)$ with domain parameter $\{a, b, g, P\}$. Each entity randomly generates φ such that $\varphi \in \mathbb{Z}_q$, computes $\phi = \varphi P$.

5.1.2 Master key setup

The ARC executes the following steps in order to generate the master key M_k for each entity.

Step 1 Using the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ as described in Lu et al. (2012), ARC chooses two random numbers $u, v \in \mathbb{Z}_p^*$ as its master key, then computes: $U_2 = g_1{}^u$; $V_2 = g_1^u$; $U_3 = g_2^u$ and then publishes the parameter $\{q, g_1, g_2, e; \mathbb{G}_1, \mathbb{G}_2, U_2, U_3, V_2, H\}$.

Step 2 For entity i, ARC randomly chooses $m_i \in \mathbb{Z}_q^*$ such that $m_i + u \neq 0 \bmod q$, and computes: $A_i = g_1 \frac{1}{m_i + u}$. It stores (i, Ai^u) in its tracking list ρ, and securely sends the authorised anonymous key, $AAK_i = (m_i, A_i)$ to the entity i.

5.1.3 SlSg keys and certificates generation stage

For each entity to generate its own anonymous authentication parameters, we modified the technique in Lu et al. (2012) by introducing a trapdoor and create life for users' key. This allows each entity to generate short-life-self-generated key and certificates. Certain number of different certificates can be generated for a key. This predefined number of certificates for a key represents the life of the key. This enhances the strength of a user's key such that even if a key is stolen, the adversary can not use it for more than the predefined lifespan before the key and its certificate expires.

Step 1 To produce a self-generated key, each entity randomly generates set $\mathbf{X}$ of order l with unique elements $x_1 \cdots x_l$ to compute another set $\mathbf{Y}$ of the same order such that $y_1 \cdots y_l \in \mathbf{Y}$, where $y_j = g_1{}^{x_j}$.

Step 2 Certificate, $cert_j$ is then generated for each of the key x_j. Each time an entity intends to use its key, the user must recompute the certificate and update the key counter *count*. Also, a case of stolen pre-computed certificates can be thwarted by updating the *count* with the time indices of the stolen certificates. A certificate is generated as follows:

For each y_j, the entity randomly chooses, $\alpha, r_\alpha, r_x, r_\delta \in \mathbb{Z}_q^*$ and computes T_u, T_v, δ, δ_1, δ_2, δ_3 $T_u = U_2^\alpha$; $T_v = V_2^\alpha$; $\delta = (\alpha \cdot x_i) \bmod q$

Then computes

$$c = H\left(U_2 \| V_2 \| y_j \| T_u \| T_v\right)$$

$$s_\alpha = r_\alpha + c \cdot \alpha \bmod q$$

$$s_x = r_x + c \cdot x_j \bmod q$$

$$s_\delta = r_\delta + c \cdot \delta \bmod q$$

$\Omega = \mu - t_n$, where μ is the life of the key, $t_1 \ldots t_{n-1}$ are the indices of all the used certificates of key x_j and t_n is the index of the current certificate.

$$\Omega_1 = \Omega^{s_\alpha};\ \Omega_2 = \Omega^{s_\delta}$$

$$\delta_1 = \frac{\Omega_1 U_2^{r_\alpha}}{\Omega_2};\ \delta_2 = \frac{\Omega_1 T_u^{r_\alpha}}{\Omega_2 U_2^{r_\delta}};\ \delta_3 = \frac{e\left(T_u,\ g_2^{r_\alpha}\right)}{e\left(V_2,\ U_3^{r_\alpha} \cdot g_2^{r_\delta}\right)}$$

$$c' = H\left(U_2 \| V_2 \| y_j \| T_u \| T_v \| \delta_1 \| \delta_2 \| \delta_3\right)$$

$cert_j = c' \| s_\alpha \| s_x \| s_\delta \| t_n$ is the valid $cert_j$ for short life self-generated key x_j at certificate count t_n.

Step 3 Obfuscation transfer of certificate-key counter: The sender transfers its certificate counter, $\Upsilon = y_j \| count \| \Omega$, to the receiver by following these obfuscation steps:

- The sender i computes $\beta = \varphi_i \phi_j$, and obfuscate its counter Υ as $F_i = \Upsilon + \beta$.
- The receiver j retrieve the sender counter $\Upsilon = F - \varphi_j \phi_i$ where $\varphi_i \phi_i$ and $\varphi_j \phi_j$ are the session key pair of the sender and receiver respectively.

5.1.4 Pseudonym generation

The certificate concatenated with y_j and Υ of the entity becomes the pseudonym F, of the entity. There is a unique pseudonym for each certificate for each session receiver. That is, $F = \{y_j \| \Upsilon \| cert_j\}$.

Table 3 Description of 2-bit cell vector

Parking space status	*Charging unit status of the space*	*Description*
0	0	Space not available, charging unit not available
1	0	Space available, charging unit not available
0	1	Space not available, charging unit available
1	1	Space available, charging unit available

6 Anonymous search for parking spaces

Whenever an AV needs a parking space, it sends its search request packet Pkt_1, as shown in Table 4, to cloud server RB. The searching procedure is summarised below.

- *Generation of the request message M:* AV generates its pseudonym for the transaction as described in the previous section. A 3-tuple search request packet Pkt_1 is then generated as:

$$Pkt_1 : F \| C_v \| d_{\max} \| \sigma$$

where $\sigma = \frac{1}{g_2^{x_j + H(Pkt_1)}}$, C_v is a 2-bit cell vector that indicates the status of the space and its charging unit as described in Table 7, and $d_{\max}$ is the maximum distance of the desired parking space to the AV. Then, Pkt_1 is securely sends to RB.

- *Verification and implementation of the request packet:* Upon receiving the securely sent packet Pkt_1, RB checks for the authenticity of the pseudonym by checking the validity of the certificate $cert_j$ in the received packet using the verification Algorithm 1.

Table 4 Search packet format

F	2-bit C_v	12-bit Maximum distance ($d_{\max}$)	σ

The algorithm generates the following decisions which RB uses to determine the authenticity and confidentiality of the received packet Pkt_1.

- *Decision 1:* Means that the certificate used in the pseudonym is either expired or invalid. That is either the key has exceeded its lifespan or the certificate is forged. In either way receiver rejects the request packet.
- *Decision 2:* This decision implies that certificate used in the pseudonym has been used before. In this case receiver rejects the request packet.
- *Decision 3:* It implies that the key is authentic and message has not been modified. In this case request packet is accepted.
- *Decision 4:* This decision confirms that the received message has been modified, therefore the request is rejected.

In a situation where the outcome is decision 3, RB gathers the status packets of all the parking lots that satisfy the requirements of the request in the format shown in Table 5. These packets are securely sent back to the AV to perform space selection and reservation in case RB is busy with other requests. Otherwise RB performs space selection, as described in the next section, and securely sends the status packets of the selected parking lot to the AV for reservation.

Table 5 Parking lot status packet format

Park-ID	Cell-ID	2-bit Cell status	12-bit cell price (Mp)	Period

7 Space selection algorithm and reservation

This section described how the propose parking management system optimises parking cost by suggesting the best parking pattern for AV. To come up with best parking pattern, we model the parking system as a transportation problem, where the objective is to assign a number of parking lots to an AV so as to minimise the total parking cost and reduce waiting time. In the model, the issue of assignment is important because of the variation in parking cost at the different parking time slots. Therefore, the problem is about how parking can be made with minimum cost and waiting time at a specific period. The

selection model is such that central cloud server RB consists of heterogenous resources (different available spaces from different PMs) to satisfy homogenous AVs.

Algorithm 1 Algorithm for certificate verification Pkt_1

Input: $\digamma$, Υ

Output: Verification Decisions I–IV

Compute $\delta_1^* = \frac{\Omega_1 U_2{}^{s_\alpha}}{\Omega_2 T_u{}^c}; \Omega^* = \mu - t_n \; \delta_2^* = \frac{\Omega_1 T_u{}^s_x}{\Omega_2 U_2{}^{s_\delta}}; \delta_3^* = \frac{e(T_{v,g_3}{}^{s_x} \cdot U_3{}^c)}{e(V_2, U_3{}^{s_\alpha} \cdot g_3{}^{s_\delta})}$

{To confirm the authenticity of the certificate}

if $(\delta_1{}^* \neq \delta_1)$ and $(\delta_2{}^* \neq \delta_2)$ and $(\delta_3{}^* \neq \delta_3)$ **then**

  Blacklist the certificate $Cert_j$;

  Reject the request of the anonymous-short-lifegenerated key $SlSgK_j$;

  return Decision 1

  {To check whether the certificate has been used before for the same anonymous-short-life-generated key $SlSgK_j$ }

else if ($t_n \in$ *certificate_counter_array*) or $(\Omega^* < 0)$ **then**

  $cert_j$ has been either used before for the same anonymous-short-life-generated key or exceeded the life span;

  return Decision 2

else

  Compute $c^* = H\,(U_2||V_2||Y_j||T_u||T_v||\delta_1{}^*||\delta_2{}^*||\delta_3{}^*)$

  {To check for the authenticity of the search packet Pkt_1}

  if $c^* = c$ **then**

    Compute $BP^* = e(Y_j \cdot g_1{}^{H(Pkt_1)}, \sigma)$

    if $BP^* = e(g_1, g_2)$ **then**

      Update the certificate-counter-array with t_n

      return Decision 3

    else

      return Decision 4

    end if

  else

    return Decision 1

  end if

end if

Let $d_{j \leftrightarrow i}$ be the distance between the current location j and next location i of an AV and β the cost per unit distance. It is assumed that an AV can park at different parking lots at different parking time slot t. Suppose AV wants to park over time T, where t is a subset of T, then it may choose to park in different parking lots during time T. This is called timepiece parking, which may reduce total parking cost over time T. In case, a parking lot satisfies minimum cost for all the considered time slot T, then an AV performs single parking for the entire time T. The goal of an AV for either single or time-piece parking is to minimise the total parking cost. The main challenge here is how to predetermine the

combination of parking spaces that gives minimum total parking cost at a reduce or no waiting time. This problem can be solved using any of the existing assignment problem solving methods such as enumeration method, simplex method, transportation method, and Hungarian method.

Hungarian method is one of the best assignment methods in the sense that it proffers the best solution without comparing every solution as in other methods (Kuhn, 1991). To use Hungarian method, a Hungarian cost matrix needs to be generated from all the available parking lots received by the AV in the search phase. Once the AV receives parking lots information from RB, it decrypts and executes the space selection procedure in phase 1 and 2 to determine the best parking pattern in case RB is busy otherwise the procedure is carried out by RB.

7.1 Phase 1

The AV generates $n \times n$ cost matrix for each time slot t, which is transformed to the Hungarian cost matrix HC by following these steps:

7.1.1 Step 1

Computation of $n \times n$ distance matrix of the parking lots as:

$$d_{j,i} = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots & d_{1,n} \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots & d_{2,n} \\ \vdots & \vdots & \vdots & \cdots & d_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{48,1} & d_{48,2} & d_{48,3} & \cdots & d_{n,n} \end{bmatrix}$$

7.1.2 Step 2

Computation of the distance cost dc, which is the cost incurred by AV driving through a distance $d_{j \leftrightarrow i}$ is computed as:

$$dc_{j \leftrightarrow i} = d_{j \leftrightarrow i} * \beta \tag{1}$$

7.1.3 Step 3

Generation of overall parking cost matrix from the distance cost matrix as:

$$CP_t(i \mapsto j) = dc(i \leftrightarrow j) + Pw_t(j \leftrightarrow j) \tag{2}$$

$$CP_t = \begin{bmatrix} CP_{t(1,1)} & CP_{t(1,2)} & CP_{t(1,3)} & \cdots & CP_{t(1,i)} \\ CP_{t(2,1)} & CP_{t(2,2)} & CP_{t(2,3)} & \cdots & CP_{t(2,i)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ CP_{t(j,1)} & CP_{t(j,2)} & CP_{t(j,3)} & \cdots & CP_{t(j,i)} \end{bmatrix}$$

where $Pw_t(j \leftrightarrow j)$ is the parking price of the parking lot j at time t as received from RB during searching.

7.1.4 Step 4

Generation of the minimum Hungarian cost matrix $\mathbb{H}_C$, such that each row of $\mathbb{H}_C$ is derived from each CP_t, and the first row of $\mathbb{H}_C$ is equal to CP_t row of the PM closest to the AV. CP_t $(i \mapsto j)$ is the parking cost for AV to park at parking lot j such that $1 \leqslant j \leqslant n$ at parking-time-window s such that $1 \leqslant s \leqslant m$, where n is total number of available parking lot and m is number of parking windows. In some cases, n and m may not be equal, we augment $\mathbb{H}_C$ to a square matrix by adding $j(n - m)$j rows.

7.2 Phase 2

Phase 2 involves generation of parking pattern for the AV by using Hungarian method to generate the minimum matrix as indicated in the following steps:

7.2.1 Step 1: location of minimum element in each row of HC

For each row $\mathbb{H}_C$, locate the lowest cost and subtract it from each cost in that row.

7.2.2 Step 2: location of minimum element in each column of HC

For each column of $\mathbb{H}_C$, locate the lowest cost and subtract it from each cost in that column.

7.2.3 Step 3

Computation of rows and columns with zeros elements and covering the respective rows and columns. Checking whether or not the number of rows and columns covered is equal to n for optimal solution otherwise step 4 is executed.

7.2.4 Step 4

Creation of additional rows and columns with zeros elements by subtracting the smallest element that is not covered from all uncovered once, and adding the element to all costs that are covered twice.

7.2.5 Step 5

Determination of parking pattern by compiling the position of the least cost in the matrix for each time slot.

7.3 Parking space reservation request

In this phase, AV sends reservation requests to all the parking lots' PMs of the selected spaces as described below. Following the same anonymous authentication procedure used during the search phase, AV composes message M_2 as: M_2: $F_{av}||\gamma||t_s$, where γ is the

parking cell identity and t_s is the timestamp. Then computes and securely sends reservation packet Pkt_2 to the selected lots as: $Pkt_2 = M_2||Ts||\sigma_{av}||y_j$.

7.4 Parking space reservation validation

The PM verifies both the certificate and message in Pkt_2. Regardless of the outcome, a reservation response packet Rt is composed by PM and securely sends back to the AV. Both the PM and AV keep the packets Pkt_2 and Rt for payment purpose. If the verification fails, PM securely sends either a response packet with status *certificate rejected CR*, *certificate expired, CE*, or *space assigned to other user*, *SA* otherwise securely sends Rt with status, *status accepted*, *SAC* in the form:

$$Rt = Pkt_2 \| F_{pm} \| F_{av} \| \sigma_{pm} \| status$$

When AV receives an acceptance reservation packet, it moves to the parking lot and submits a copy of the Rt to the PM for certificate and signature verification. The likely cause of conflict in this phase is when either repudiation occurred or remodification detected by either AV or PM. These instances are handled by ARC by verifying the authenticity of the certificate and signature on the packet. Then uses its tracking list to identify the real identity of the owner of the controversial packet.

8 Evaluation and security analysis

8.1 Security analysis

This section contains the formal security analysis of the proposed parking system against all the threats described in the threat model. The security and functionality of SAPMS depend solely on the following:

- One way hash function H which is used to generate signature.
- ECDLP that is introduced in the session key generation and obfuscation transfer stages.
- Discrete log problem (DLP) introduced in short-life-self generated key, certificate and pseudonym generation.

8.1.1 Elevation of privilege, message modification and repudiation attacks through signature forging

In the propose parking system, repudiation and message modification attacks are thwarted by the signature on the packet The signature $\sigma = \frac{1}{g_2^{x_j + H(Pkt_1)}}$ is an integral part of the transmitted packet. The signature consists of the sender key x_j and hash of the message, which are respectively protected by DLP and non-collision property of hash function. Inasmuch definition 1 and 2 in the primitive section hold, signature can not be compromised. This assumption therefore makes it easy for verifier to thwart either

elevation of privilege, message modification or repudiation attack launch through signature forging. To thwart these attacks, verifier performs the following check:

$$e\left(Y_j \cdot g_1^{H(Pkt_1)}, \sigma\right) \stackrel{?}{=} e\left(g_1, g_2\right)$$

If it holds, then the signature verified. That is, for a packet Pkt_1 signed as $\sigma = \frac{1}{g_2^{x_j + H(Pkt_1)}}$, if an adversary A, who intention is to modify Pkt_1 and generate σ' such that $\sigma' = \sigma$ then:

$$A_\sigma\left(Pkt_1\right) = \Pr\left[\left(Pkt_1, Pkt_1'\right) \Leftarrow RA : Pkt_1 \neq Pkt_1', \sigma\left(Pkt_1\right) = \sigma\left(Pkt_1'\right)\right]$$

where $Pr(L)$ denotes the probability of a random attack on message Pkt_1 through signature σ, and $(Pkt_1, Pkt_1') \Leftarrow RA$ denotes the pair (Pkt_1, Pkt_1') is selected randomly by A. If the adversary success rate in finding collision for $A_\sigma\ (Pkt_1) \leq 0$, then hash function is collision resistant and DLP is computationally difficult. Therefore, the scheme is secured against any message authentication related attacks such as repudiation and modification, etc.

Elevation of privilege or sybil attack can be launched through stolen or reused certificate. The verification algorithm is capable of detecting any attempt to launch this attack either through the use of stolen or modified certificate.

The verifier received pseudonym $F = Y_j||T_u||T_v||c'||s_\alpha||s_x||s_\delta$. To confirm whether the pseudonym is generated from either a stolen certificate, expired key or stolen key, the verifier perform the following:

- Computes

$$\delta_1^* = \frac{\Omega_1 U_2^{s_\alpha}}{\Omega_2 T_u^{\,c}}; \delta_2^* = \frac{\Omega_1 T_{ux}^{\,s}}{\Omega_2 \delta^*{}_1 U_2^{s_\delta}}; \delta_3^* = \frac{e\left(T_{v,g_2}{}^{s_x} \cdot U_3{}^c\right)}{e\left(V_2, U_3{}^{s_\alpha} \cdot g_2{}^{s_\delta}\right)}$$

$$c' = H\left(U_2 \| V_2 \| y_j \| T_u \| T_v \| \delta_1^* \| \delta_2^* \| \delta_3^*\right)$$

- Compares if the newly computed c^* is equal to the c' embellished in the pseudonym. If they are no equal, it shows that either the Ω_1 or Ω_2 had been maliciously modified then elevation of privilege detected. However, in case of stolen or reused certificate, c^* is equal to the c'. The verifier needs to go further by checking the composition of the two Ω_1 or Ω_2, since Ω_1 and Ω_2 are generated from $\Omega = \mu - t_n$ where all the used t and μ are in the received *count* obfuscately received from sender. The verifier then compute set of $\delta^*{}_1$ for each t in the count if anyone of them is equal to the pseudonym δ_1, then the pseudonym was generated from either a stolen certificate, expired key or used certificate.

The above formal analysis depends on the *count* that is obfuscately exchanged between sender and receiver of the packet. The obfuscate transfer depends on the ECDLP problems as described in Definition 3. Inasmuch the problem is computationally difficult the count can not be compromised, the certificate authentication remains valid, thus the system is secured.

8.1.2 Spoofing on identity

Identity of entity can not be spoof upon by adversary except it collude with ARC, who we assume is a trusted party. Only ARC has a tracking list ρ, which consists of real identities and corresponding master keys A of entities. Inasmuch this is secured or not compromised, no adversary can spoof on identity. This can be formally described as: The master key A_i is blinded during certificate generation as $T_v = A_i V_2$, where $V_2 = g_1^v$. This operation involves DLP, which has been shown to be difficulty to solve. Inasmuch this problem is computationally difficult, as described in Definition 2, and ARC does not compromised the tracking list, adversary can not link the identity with pseudonym.

8.1.3 Information disclosure and DoS

The use of session key to secure the packet transfer, through asymmetric encryption, protect information exchange in the SAPMS. The difficulty in breaking the encrypted packet lies on the ECDLP. For DoS attacks on RB, the parking management system allows AV to perform space selection in case RB is overwhelmed by requests. Decentralisation of the space selection make DoS attack impossible on the RB except RB is compromised and intentionally refused to take requests from AV.

8.2 Computation cost analysis

In this section, the computational cost of the anonymous authentication scheme of SAPMS was analysed and compared against two state-of-the-art (SoA) anonymous authentication schemes proposed in Lu et al. (2008) and Jung et al. (2009). The corresponding computation time of the cryptography operations used in the three schemes obtained from Jung et al. (2009), as shown in Table 6, are used to determine the overall computation time for each scheme. Table 7 shows the computation cost of the two anonymous authentication scheme in Jung et al. (2009) and estimated computation cost of our propose SAPMS' anonymous authentication scheme. The results show that our authentication scheme is the best in terms of computation speed compare to the schemes in Lu et al. (2008) and Jung et al. (2009).

Table 6 Cryptographic operation time (implemented on Pentium IV 3.0 GHz)

Cryptographic operation time	*Time (ms)*
$\hat{e}$ bilinear pairing operation	4.5
Point multiplication on G1	0.6
Exponentiation on $\mathbb{Z}_q$	2.1

Table 7 Computation cost analysis in (ms)

Operation	*ECPP in Lu et al. (2008)*	*Anonymous scheme in Jung et al. (2009)*	*Propose scheme*
n certificate issue	20.4 + 14.4n	12.6 + 18.6 n	0.28 + 12.06 n
n certificate verification	17.1 n	17.1 n	0.28 + 12.06 n
Encryption and decryption	No need	No need	2n
Session key generation	No need	No need	1.2n
Total time (ms) for n = 1	51.9	48.3	29.08

8.2.1 Evaluation of the space selection and reservation

The proposed cost optimisation selection and reservation part of our parking system are simulated using real parking lots information obtained from the downtown Nashville parking options. The performance of the propose cost optimisation selection and reservation was demonstrated in terms of the correctness of the output of the optimal cost solutions. Two different searches at different locations were carried out to generate our test samples. A standard mile per gallon value β of 57.5 cents was used to generate the distance cost matrix dc from distance matrix d, and cost matrix $\mathbb{H}_C$. Four parking-time slots were used (t_1 = 12 am – 5 am; t_2 = 6 am – 6 pm; t_3 = 6 pm – 6 am; t_4 = 24 hours).

In the first evaluation test, parking lot information was returned for six parking lots in the search phase containing their corresponding prices and distance. These were used to generate matrices d, $\mathbb{P}w$ and dc.

$$d = \begin{bmatrix} 0 & 0.8 & 1.3 & 0.1 & 1.2 & 0.3 \\ 0.8 & 0 & 1.1 & 1.1 & 0.6 & 0.9 \\ 0.9 & 1.1 & 0 & 0.9 & 0.7 & 0.9 \\ 0.1 & 1.1 & 0.9 & 0 & 0.7 & 0.5 \\ 1.2 & 0.6 & 0.7 & 0.7 & 0 & 0.5 \\ 0.3 & 0.9 & 0.9 & 0.5 & 0.5 & 0 \end{bmatrix}$$

$$dc = \begin{bmatrix} 0 & 0.46 & 0.42 & 0.06 & 0.69 & 0.17 \\ 0.46 & 0 & 0.63 & 0.63 & 0.35 & 0.52 \\ 0.52 & 0.63 & 0 & 0.52 & 0.40 & 0.52 \\ 0.06 & 0.63 & 0.52 & 0 & 0.40 & 0.29 \\ 0.69 & 0.35 & 0.40 & 0.40 & 0 & 0.29 \\ 0.17 & 0.52 & 0.52 & 0.29 & 0.29 & 0 \end{bmatrix}$$

Using an index of 1 (meaning the AV is at PM_1), the Hungarian cost matrix $\mathbb{H}_{C1}$ was generated as:

$$\mathbb{H}_{C1} = \begin{bmatrix} 10.456 & 3.000 & 5.627 & 8.627 & 20.342 & 18.513 \\ 12.456 & 5.000 & 8.627 & 8.627 & 25.342 & 18.513 \\ 10.456 & 5.000 & 8.627 & 8.627 & 25.342 & 18.513 \\ 22.456 & 8.000 & 8.627 & 10.627 & 25.342 & 24.513 \end{bmatrix}$$

The resulting parking pattern showing the parking lot that gives the minimum parking cost for parking slots $t_1 \cdots t_4$ are shown in Figures 2 and 3. Figure 2 shows the parking pattern for single parking mode while Figure 3 shows the pattern for time piece parking mode. In both figures each row represents the parking cost at each time slot. The shaded cells indicate the minimum cost for each time slot.

Figure 2 Single parking mode (see online version for colours)

10.456	3.000	5.627	8.627	20.342	18.513
12.456	5.000	8.627	8.627	15.342	18.513
10.456	5.000	8.627	8.627	15.342	18.513
22.456	8.000	8.627	10.627	15.342	24.513

Figure 3 Time-piece parking mode (see online version for colours)

10.456	**3.000**	5.627	8.627	20.342	18.513
12.456	5.000	8.627	**8.627**	15.342	18.513
10.456	5.000	8.627	8.627	15.342	18.513
22.456	8.000	**8.627**	10.627	15.342	24.513

Figure 4 Single parking mode (see online version for colours)

7.46	9.00	9.63	10.63	11.35	18.52
12.46	**5.00**	8.63	8.63	15.35	18.52
8.46	6.00	8.63	**4.63**	12.35	16.52
16.46	16.00	17.63	18.63	**14.35**	20.12

Figure 5 Time-piece parking mode (see online version for colours)

7.46	9.00	9.63	10.63	11.35	18.52
12.46	**5.00**	8.63	8.63	15.35	18.52
8.46	6.00	8.63	**4.63**	12.35	16.52
16.46	16.00	17.63	18.63	**14.35**	20.12

In the second evaluation test, the location of the AV was changed from PM_1 to PM_2 and a new request was sent to the RB. A new set of parking lots was obtained, and new $\mathbb{CP}_{t2}$ was generated as

$$\mathbb{CP}_{t2} = \begin{bmatrix} 7 & 9 & 9 & 10 & 11 & 18 \\ 12 & 5 & 8 & 8 & 15 & 18 \\ 8 & 6 & 8 & 4 & 12 & 16 \\ 16 & 16 & 17 & 18 & 14 & 20 \end{bmatrix}$$

The new Hungarian cost matrix $\mathbb{H}_{C1}$ was generated using index 2 as:

$$\mathbb{H}_{C2} = \begin{bmatrix} 7.460 & 9.000 & 9.630 & 10.630 & 11.350 & 18.520 \\ 12.460 & 5.000 & 8.630 & 8.630 & 15.350 & 18.520 \\ 8.460 & 6.000 & 8.630 & 4.630 & 15.350 & 16.520 \\ 16.460 & 16.000 & 17.630 & 18.630 & 14.350 & 20.520 \end{bmatrix}$$

The resulting parking pattern for the second evaluation test are as shown in Figure 4 and 5. The Hungarian method was applied on $\mathbb{H}_{C2}$ and the optimal cost for both single and time-piece parking patterns were obtained: The results of the evaluation test I as depicted in Figure 2 and 3 showed that the minimum parking cost for the entire time $t_1 \cdots t_4$ for single parking mode was \$21:00 while the minimum parking cost for time-piece parking mode was \$30:71. This indicates that the minimum parking cost was obtained when the AV adopted single parking mode for the entire parking slots $t_1 \cdots t_4$. This confirms that single parking mode is cheaper than time-piece parking mode for the period under consideration. However, this is subject to the availability of the selected parking lots at the time of reservation. In the second evaluation test, the results indicated that a minimum cost of \$31:44 was obtained in time-piece parking mode for the entire time period $t_1 \cdots t_4$ as against \$36 obtained for single parking mode for the same period of time. During the period of the second evaluation test, it is cheaper for the AV to adopt time-piece parking rather single parking mode.

9 Conclusions

In this work, we have proposed a secure and anonymous parking management system for level-5 vehicles using a novel space selection technique to reduce parking cost and waiting time. The main idea is to allow AV to perform timepiece parking, anonymous authentication, and decentralise the generation of authentication keys and certificates. Compared the anonymous authentication scheme of the system with SoA anonymous scheme in terms of computational cost, the system's anonymous authentication scheme computational cost is half of the SoA. The space selection technique was able to select parking pattern with lowest cost.

References

Au, M.H., Liu, J.K., Fang, J., Jiang, Z.L. and Susil, W. (2014) 'A new payment system for enhancing location privacy of electric vehicles', *IEEE Transactions on Vehicular Technology*, Vol. 63, No. 1.

Biondi, S., Monteleone, S. and Catania, V. (2016) 'A context-aware smart parking system', *12th International Conference on Signal-Image Technology & Internet-Based Systems*.

Boukerche, A., Oliveira, H.A.B.F., Nakamura, E.F. and Loureiro, A.A.F. (2008) 'Vehicular ad hoc networks: a new challenge for localization-based systems', *Computer Communication*.

Chim, T.W., Cheung, J.C.L., Yiu, S.M., Hui, L.C.K. and Li, V.O.K. (2011) 'SPCS: secure and privacy- preserving charging-station searching using VANET', *Journal of Information Security*, Vol. 3, No. 1, pp.59–67.

Cho, W., Park, Y., Sur, C. and Rhee, K.H. (2013) 'An improved privacy preserving navigation protocol in VANETs', *Journal of Wireless Mobile Networks, Ubquitous Computing and Dependable Applications*, Vol. 4, No. 4, pp.80–92.

Ellison, C. and Schneier, B. (2000) 'Ten risks of PKI: what you're not being told about public key ten risks of PKI: what you're not being told about public key infrastructure', *Computer Security Journal*, Vol. 16, No 1, pp.1–8.

Jung, C.D., Sur, C., Park, Y. and Rhee, K-H. (2009) 'A robust and efcient anonymous authentication protocol in VANETs', *Journal of Communications and Networks*, Vol. 11, No. 6, pp.607–614.

Kim, T.H-J., Studer, A., Dubey, R., Zhang, X., Perrig, A., Bai, F., Bellur, B. and Iyer, A. (2010) 'Vanet alert endorsement using multi-source filters', *VANET' 10*.

Kokalj-Filipovc, S. and Fessant, F.L. (2010) 'Personal social graph as an anonymous vehicle for P2P applications: the cost of renting trusted connections', *44th Annual Conference on Information Sciences and Systems*, pp.1–6.

Kuhn, H.W. (1991) '*On the Origin of the Hungarian Method, History of Mathematical Programming; A Collection of Personal Reminiscences*, in Lenstra, J.JK., Rinnooy Kan, A.H.G. and Schrijver, A. (Eds.): pp.77–81, North Holland, Amsterdam.

Li, H., Dan, G. and Nahrstedt, K. (2015a) 'Lynx: authenticated anonymous real-time reporting of electric vehicle information', *IEEE International Conference on Smart Grid Communications (SmartGrid-Comm)*, Miami, FL, 2015, pp.599–604.

Lochert, C., Sceuermann, B. and Wewetzer, C., Luebke, A. and Mauve, M. (2008) 'Data aggregation and roadside unit placement for a VANET traffic information system', *Proceeding of VANET '08 the Fifth ACM International Workshop on Vehicular Inter-Networking*, pp.58–65.

Lu, R., Lin, X., Luan, T., Liang, X. and Sherman, X. (2012) 'Pseudonymn changing at social spots: an effective strategy for location privacy in VANETs', *IEEE Transactions on Vehicle Technology*, Vol. 61, No. 1, pp.86–96.

Lu, R., Lin, X., Zhu, H. and Shen, X. (2009) 'SPARK: a new VANET-based Smart parking scheme for large parking lots', *Proceedings of IEEE INFOCOM*.

Lu, R., Lin, X., Zhu, H., Ho, P-H. and Shen, X. (2008) 'ECPP: efficient conditional privacy preservation protocol for secure vehicular communications', in *Proc. IEEE INFOCOM*, p.19031911.

Mahmud, S.A., Khan, G.M., Rahman, M. and Zafar, H. (2013) 'A survey of intelligent car parking system', *Journal of Applied Research and Technology*, Vol. 11, No. 5, pp.714–726.

Microsoft (2005) [online] *The STRIDE Threat Model*, *Microsoft* [online] https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20)aspx (accessed 15 February 2016).

Mustafa, M.A. and Zhang, N. (2014) 'Roaming electric vehicle charging and billing: an anonymous multi-user protocol', *IEEE International Conference on Smart Grid Communications*.

Zadeh, N.R.N. and Dela, J.C. (2016) 'Smart urban parking detection system', *6th IEEE International Conference on Control System, Computing and Engineering*, pp.370–373.

Zhao, T., Chen, C., Wei, L. and Yu, M. (2014) 'An anonymous payment system to protect the privacy of electric vehicles', *Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*.