Portunes: Privacy-Preserving Fast Authentication for Dynamic Electric Vehicle Charging

Hongyang Li
University of Illinois Urbana-Champaign
hli52@illinois.edu

György Dán
KTH Royal Institute of Technology
gyuri@kth.se

Klara Nahrstedt
University of Illinois Urbana-Champaign
klara@illinois.edu

***Abstract*—Dynamic contactless charging is an emerging technology for charging electric vehicles (EV) on the move. For efficient charging and for proper billing, dynamic charging requires secure communication between the charging infrastructure and the EVs that supports very frequent real-time message exchange for EV authentication. In this paper we propose *Portunes*, an authentication protocol for charging pads to authenticate an EV's identity. *Portunes* uses pseudonyms to provide location privacy, allows EVs to roam between different charging sections and receive a single bill, and achieves fast authentication by relying on symmetric keys and on the spatio-temporal location of the EV. We have implemented *Portunes* on RaspberryPi Model B with 700 MHz CPU and 512 MB RAM. *Portunes* allows the EV to generate authentication information within 0.3 ms, and allows charging pads to verify the information within 0.5 ms. In comparison, ECDSA signature generation and verification take over 25 ms and over 40 ms respectively.**

I. Introduction

Dynamic contactless charging [1] is a promising technology for charging electric vehicles (EV) while they drive and has attracted attention from both the industry [2] and the research community [3]–[5]. In the case of dynamic contactless charging, charging pads are placed under a contiguous section of roadbed, called a charging section, and the EV's battery is charged through magnetic induction between the coils in the charging pads under the road and coils installed at the bottom of the EV as the EV passes over the pads.

In order to allow a significant amount of energy to be transmitted to an EV in a charging section, a charging section has to be several kilometers long. At the same time, to facilitate power management and to be able to maintain an operable voltage along the charging section, a charging section has to consist of many small charging pads (e.g., 40 cm-long charging pads) rather than being a single long charging pad under the road. Each charging pad can be switched on and off individually, and should only be switched on when there is an EV above it for two reasons: to reduce energy consumption and to avoid electromagnetic radiation of humans and animals that could potentially enter the charging section. The charging rate and resonance frequency of each charging pad can also be controlled individually, thus the charging section can simultaneously charge multiple EVs with different battery types and coils.

While the micro charging pad approach facilitates power management, it requires communication between the EVs and the pads. First, the EV needs to inform each micro charging pad about its arrival just in time for the pad to switch on, and about its charging parameters, such as the desired charging rate, battery type, coil type, etc. Second, the charging pad must be able to authenticate the incoming EV in order to bill the correct customer. If the EV is moving at high speed (e.g., 100 km/h), the contact time between the EV and a charging pad might be only tens of milliseconds.

Since there are many short charging pads in a section, dynamic charging requires high authentication frequency, and thus the authentication protocol has to be fast and lightweight. Verifying a digital signature could take tens of milliseconds [6] and is infeasible in this scenario. One-time signature schemes [7], [8] that feature fast signature verification come at the cost of slow key generation or large key size, and thus cannot achieve fast mutual authentication. Authentication based on challenge-response [9] that requires multiple message exchanges is less likely to succeed due to packet losses in vehicular networks [10].

In this paper we propose *Portunes*, an authentication protocol that provides location privacy through using pseudonyms, allows EVs to roam between different charging sections and receive a single bill, and achieves fast authentication by relying on symmetric keys and on the spatio-temporal location of the EV. To strike the right balance between computational cost and authentication security and efficiency, *Portunes* adopts a key pre-distribution approach. Efficient key pre-distribution is enabled by the heavy daily fluctuation of road traffic: a road can be crowded during rush hour, but can be nearly empty during night time. *Portunes* utilizes the periods when there is little road traffic to generate and to pre-distribute session keys to the charging pads, so that an EV can obtain and use a session key with the charging pads even during rush hours without having to dimension the communication capacity of the charging pads for peak hours.

The rest of the paper is organized as follows: in Section II we describe the system model; in Section III we present the Portunes protocol; in Section IV we analyze various security aspects of *Portunes*; in Section V we discuss several related issues; in Section VI we present evaluation results; in Section VII we review related work, and conclude our paper in Section VIII.

II. Model and Assumptions

We consider a system that consists of *charging service providers* (CSP), *pad owners* (PO) and electric vehicles (EVs).

a) Physical Model: We assume charging pads are deployed sequentially under the roadbed in the charging section. The length of the charging section could be in the order of kilometers. We denote the length of a charging pad by λ, and

the distance between two charging pads by δ. A typical setup might be $L = 4$ km, $\lambda = \delta = 0.4$ m.

b) Communication Model: We assume that the CSP and the PO are connected through a high speed IP network. We make the reasonable assumption that the PO will communicate with its charging pads via power-line communication (PLC), as this keeps the roadbed infrastructure simple. PLC is able to meet the bandwidth requirement since periods of low traffic typically last for several hours, during which time the PO can transmit key materials for the next day to each charging pad. Finally, each EV can communicate with the CSP either via the cellular network or via WiFi through roadside units (RSU).

For EV to charging pad communication, we consider that there is a dedicated short range wireless communication device installed at the bottom of the EVs; we denote its vertical distance from the ground by h. A corresponding short range wireless communication device is installed at the beginning of each pad. We denote the range of the wireless device by r, and denote the communication contact time between the EV and a pad, which is defined as the duration when the EV and the pad can communicate with each other, by T.

A typical setup might be $r = 0.5$ m, and $h = 0.3$ m. Note that in this case the wireless devices at two neighboring pads are separated by $\lambda + \delta = 0.8$ m, and at most one charging pad will receive the transmitted signal from an EV. Due to the short communication range, a pad is also unlikely to receive the beacon from an EV moving at another lane. If the EV is moving at speed $v = 108$ km/s then the communication contact time $T = (\frac{2\sqrt{r^2-h^2}}{v} =) 20$ ms.

c) Time and Location Information: We assume that the CSP, the PO, each EV, and each charging pad all have a clock with time accuracy no worse than 200 ms. An EV can synchronize its clock with either GPS satellite if it has on-board GPS device, or with an Internet time server through WiFi or cellular connection. Most Real Time Clocks (RTC) commonly used in electronic devices today can achieve an accuracy of around 100 ppm (1 parts-per-million (ppm) = 10^{-6}), and an EV using such RTC only needs to synchronize its clock every ($\frac{200 \text{ ms}}{100 \text{ ppm}} =$) 33 mins. Each charging pad p synchronizes with the PO's clock using some network clock synchronization algorithm (e.g., [11]), and learns its GPS coordinates l_p from the PO.

d) Billing Model: Each PO maintains a charging section to provide dynamic contactless charging service to EVs. The PO does not bill the EV directly for the service, but it bills the CSP with which the EV has a contract with. The CSP pays the PO for charging its subscribing EVs, and then bills the EV. Each CSP may have service agreements with multiple POs, and an EV is free to choose which CSP it subscribes to. This roaming billing model is inspired by the way mobile phone billing for roaming works today.

e) Security and Attack Model: We assume deployment of a PKI. The CSPs, the POs, and each EV have a pair of public and private keys. Each CSP knows the subscribing EVs' public keys, and each EV also knows its CSP's public key. The CSPs know the public keys of the POs and vice-versa. In addition, a PO P shares a symmetric key $K_{P,p}$ with each of its charging pads p. Each CSP C also shares a one-way function $f_{C,P}$ with each PO P (and its charging pads).

TABLE I: Notations

I_e	the permanent identity of EV e.
π	pseudonym assigned by the CSP to an EV.
Π	the set of all pseudonyms.
$f_{C,P}$ (or f)	collision-free one-way function shared between CSP C and PO P.
$K_{f(\pi)}$	session key assigned by the CSP to EV with pseudonym π.
$\mathcal{K}_{C,P}$ (or $\mathcal{K}$)	the key set $\{(\pi, K_{f(\pi)}) : \pi \in \Pi\}$ of all index-key pairs sent by CSP C to PO P.
$K_{P,p}$	symmetric key shared between pad p and PO P.
$K(m)$	AES encryption of message m using symmetric key K.
$\{m\}_{A \to B}$	asymmetrically encrypt (e.g., RSA) the message m using B's public key, then sign the encrypted message using A's private key.
t_A	timestamp generated by A.
$\hat{l}_e(t)$	the estimated location of EV e at time t.
$l_e(t)$	the true location of EV e at time t.
l_p	the true location of charging pad p
ϵ_l	acceptable error in the location stamp.
ϵ_t	acceptable error in the time stamp.
r	communication range of the wireless devices installed at the bottom of each EV and at the start of each pad.
h	vertical distance from the wireless device at the bottom of the EV to the ground.

We assume the attacker is computationally bounded, i.e., the attacker cannot reverse a one-way function or crack an AES encryption using brute force. We assume the attacker cannot compromise the CSP, the pad owner, or any charging pad.

In Table I we summarize the notations used in the paper. To simplify notations, we use f and $\mathcal{K}$ to denote $f_{C,P}$ and $\mathcal{K}_{C,P}$ respectively when C and P are clear from the context.

III. Portunes

Portunes aims to provide simple, robust, scalable, and privacy-preserving authentication, but not to optimize the charging process itself. Operational and control issues such as choosing the optimal charging rate, scheduling when to switch on and off each charging pad, accounting for inefficient charging when the charging coils are not properly aligned (e.g., when the EV is switching lanes), are beyond our scope.

Portunes consists of three major phases: (i) key pre-distribution; (ii) authentication; and (iii) accounting. In the key pre-distribution phase, the CSPs generate the key sets and send them to the POs, which in turn disseminate the key sets to each charging pad. In the authentication step, the CSPs allocate keys and pseudonyms to EVs before they enter the charging section, and the EVs authenticate with each charging pad encountered using the assigned key. The true identity of the EV is not revealed to the charging pads during the authentication. In the final accounting phase, the PO collects information from the charging pads, calculates the total amount of energy provided to each EV, and sends this information to the CSPs. Note that if the EV does not want to use dynamic charging at all, it does not send any message to the CSP or to any charging pads.

In Fig. 1 we show the message exchange; for simplicity we only show the message exchange between an EV e and a single charging pad p.

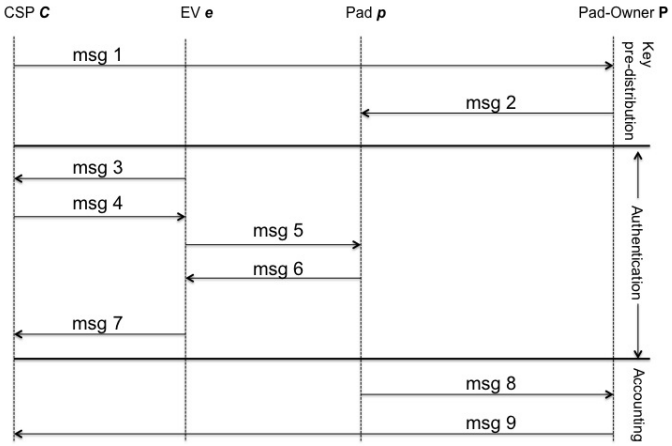

Fig. 1: *Portunes* protocol overview with CSP C, EV e, pad-owner P and charging pad p. Messages 1 to 9 are specified in equations (1)-(9), respectively.

A. Key Pre-distribution Phase

The key pre-distribution phase occurs every night, when there is little road traffic. CSP C generates the pseudonym set Π and the corresponding indexed key set $\mathcal{K} = \{(f(\pi), K_{f(\pi)}) : \pi \in \Pi\}$ using a collision-free one-way function f, where $K_{f(\pi)}$ is a session key with index $f(\pi)$. f is one-way in that it is infeasible to compute π given $f(\pi)$. Since we assume the pseydonym set Π and the key set $\mathcal{K}$ are generated daily, the size of Π and $\mathcal{K}$ depends on the daily traffic volume at the charging section [1]. For each $\pi \in \Pi$, CSP C sends

$$\text{msg } 1 : \{K_{f(\pi)}, f(\pi), t_C\}_{C \to P} \tag{1}$$

to the PO, where t_C is a timestamp generated by C. Note that msg 1 is asymmetrically encrypted using P's public key so that only P can decrypt the message (using its private key), and signed by C's private key to ensure its authenticity.

When receiving msg 1, the PO disseminates the learned index-key pairs $(f(\pi), K_{f(\pi)})$ to the charging pads by sending the message

$$\text{msg } 2 : K_{P,p}(f(\pi), K_{f(\pi)}, t_P) \tag{2}$$

to each pad p, where t_P is the current timestamp generated by the PO. This message is encrypted using AES with key $K_{P,p}$, which is only shared between PO P and pad p. In the end, each charging pad learns the entire key set $\mathcal{K}$.

B. Authentication with CSP and Charging Pads

Upon entering a charging section, EV e authenticates with CSP C to obtain a pseudonym π and the key $K_{f(\pi)}$. As the EV moves within the charging section, it uses π and $K_{f(\pi)}$ to authenticate with each charging pad it encounters.

[1]The annual average daily traffic (AADT) of highly congested road is generally in the order of hundreds of thousands cars. This implies that in the extreme case where every EV in a congested road requires dynamic charging, the size of Π is at most some hundreds of thousands.

1) EV-CSP Authentication: In order to authenticate with the CSP, EV e sends

$$\text{msg } 3 : \{I_e, t_e\}_{e \to C} \tag{3}$$

to CSP C upon entering the charging section. Here I_e is the permanent ID of EV e, and t_e is a timestamp generated by EV e. The EV encrypts the plaintext message using CSP's public key, and then signs the message.

When receiving msg 3 from EV e, CSP C verifies the digital signature, and decrypts the message using its private key. It also verifies that the timestamp t_e is within a valid range. C then selects an unassigned pseudonym $\pi \in \Pi$ at random and sends

$$\text{msg } 4 : \{I_e, t_e, t_C, \pi, K_{f(\pi)}\}_{C \to e} \tag{4}$$

back to EV e, where $K_{f(\pi)}$ is the session key corresponding to π, t_e is the timestamp received in msg 3, and t_C is the CSP's current time. Note that only EV e can decrypt msg 4 since it is encrypted using e's public key. The message is also signed by CSP C to ensure its authenticity.

2) EV-Pad Authentication: Once on the charging section, in order to authenticate with a charging pad within range, EV e periodically broadcasts the beacon

$$\text{msg } 5 : beacon = (\pi, K_{f(\pi)}(C, \pi, t_e, \hat{l}_e(t_e), req)), \tag{5}$$

where C is the CSP that assigned the pseudonym π and the session key $K_{f(\pi)}$ to EV e, t_e is the current timestamp generated by the EV, and $\hat{l}_e(t_e)$ is the estimated location of EV e at time t_e. The req field contains charging parameters needed by the charging pad, such as the EV's battery and coil type and the desired charging rate. The broadcast frequency is determined by the EV based on its speed. As an example, if pads are $\lambda = 0.4$ m long and are spaced $\delta = 0.4$ m, an EV moving at 108km/h may broadcast the beacon every 5 ms.

The pseudonym π in plaintext is used by the pad to locate the corresponding session key $K_{f(\pi)}$. When pad p receives the beacon, it uses the mapping f shared with CSP C to compute $f(\pi)$, and uses key $K_{f(\pi)}$ to decrypt the cipher text. Pad p verifies that: (i) the plaintext and the encrypted pseudonyms match; (ii) t_e is valid, by checking $|t_e - t_p| < \epsilon_t$, where ϵ_t is the accepted time mismatch; and (iii) $\hat{l}_e(t_e)$ is valid, by checking $\|\hat{l}_e(t_e) - l_p\| < \epsilon_l$, where ϵ_l is the accepted location mismatch. We discuss how to determine the values of ϵ_l and ϵ_t in Section III-D and IV-A, respectively.

If all verifications succeed then pad p will switch on and charge the EV. At the same time it removes the corresponding key $K_{f(\pi)}$ from its local storage, and thus ignores any further messages using the same pseudonym π.

If verifications (i) and (ii) succeed (i.e., whether or not the EV's estimated location $\hat{l}_e(t_e)$ is accurate enough), pad p sends

$$\text{msg } 6 : K_{f(\pi)}(\pi, t_e, t_p, l_p, ack) \tag{6}$$

to EV e, where t_e is the timestamp received in EV e's beacon, t_p is the timestamp generated by pad p, and l_p is the pad's known location. The ack field contains semantic information

for the EV, such as whether the EV is properly aligned with the charging pad, or whether the EV should adjust its speed.

If the location estimate $\hat{l}_e(t_e)$ is inaccurate (and thus verification (iii) fails) then the pad will not switch on, but it will still send msg 6 to the EV. In this case the l_p field in msg 6 helps EV e to improve its location estimate. Note that even if msg 6 is lost, the charging pad would still charge the EV if it has received an authentic msg 5 from the EV.

Finally, before EV e leaves the charging section, it reports to CSP C the amount of energy E received in the section

$$\text{msg } 7 : \{I_e, t_e, E\}_{e \to C} \tag{7}$$

The message is encrypted using CSP C's public key, and then digitally signed using EV e's private key.

C. Accounting Phase

The accounting phase is performed during times of little road traffic. Each pad p sends the following message to PO P

$$\text{msg } 8 : K_{P,p}(p, \pi, t^0_{p,\pi}, t^1_{p,\pi}, E_{p,\pi}), \tag{8}$$

where π is an EV's pseudonym, $t^0_{p,\pi}$ and $t^1_{p,\pi}$ are the start and end time that pad p charged the EV, and $E_{p,\pi}$ is the amount of energy the pad supplied to the EV with pseudonym π.

The PO calculates the total supplied energy $E_\pi = \sum_p E_{p,\pi}$, as well as the first time t^0_π and the last time t^1_π that the EV with pseudonym π used its dynamic charging service. The timestamps are useful for accurate dynamic pricing. PO P then sends the following message to CSP C

$$\text{msg } 9 : \{\pi, t^0_\pi, t^1_\pi, E_\pi\}_{P \to C} \tag{9}$$

The financial settlement is based on E_π reported by the PO.

D. Estimating the EV's location

Recall that in order for a pad to switch on, *Portunes* requires that EV e's location estimate $\hat{l}_e(t)$ be within ϵ_l of its actual location. The simplest solution for an EV to estimate its location would be to use its on-board GPS, but this solution has several drawbacks. First, the horizontal accuracy of GPS is up to 2.2 meters with 95% probability [12], thus the range may include the locations of multiple charging pads. Second, GPS signals may be unavailable, e.g., in tunnels. Third, a failure of the GPS receiver would prevent an EV from using dynamic charging, hence from reaching its destination. We argue that such a dependency on a built in system would be undesirable.

It is for these reasons that *Portunes* assists the EV's location estimation through including l_p in msg 6. Note that if EV e is able to receive msg 6 at time t from pad p, then the horizontal distance $\|l_p - l_e(t)\|$ between pad p and EV e at time t satisfies

$$\|l_p - l_e(t)\| < \sqrt{r^2 - h^2} + \bar{v} \cdot \tau, \tag{10}$$

where $\sqrt{r^2 - h^2}$ is the maximum horizontal distance between the wireless device at the bottom of the EV and the charging pad in its communication range r, τ is the transmission delay of msg 6, and $\bar{v}$ is the EV's average speed during time τ.

In *Portunes* if EV e receives msg 6 from pad p then it updates its estimated location $\hat{l}_e(t)$ to pad p's location l_p, as this provides very good accuracy. As an example, if $r = 0.5$ m, $h = 0.3$ m, $\bar{v} = 108$ km/h, and $\tau = 1$ ms, the location estimation error is $\|\hat{l}_e(t) - l_e(t)\| = \|l_p - l_e(t)\| < 0.45$ m, which is significantly less than GPS's horizontal accuracy of 2.2 m at 95% confidence.

Upon sending the next beacon at time t', the EV can estimate its location

$$\hat{l}_e(t') = \hat{l}_e(t) + \vec{v}_e(t) \cdot (t' - t) \tag{11}$$

where t is the last time that EV e receives msg 6 from some pad p, $\vec{v}_e(t)$ is the EV's velocity at time t, and $\hat{l}_e(t)$ is the EV's location estimation at time t when it receives msg 6 from pad p, i.e., $\hat{l}_e(t) = l_p$. If an EV broadcasts a beacon every few milliseconds, $t' - t$ is small, and the EV's velocity change during (t, t') can be neglected.[2]

In order for pad p to receive a beacon from EV e, their horizontal distance must be less than $\sqrt{r^2 - h^2}$. Therefore, the allowed location error ϵ_l must satisfy $\epsilon_l > \sqrt{r^2 - h^2} + \|\hat{l}_e(t) - l_e(t)\|$. In our example where $\|\hat{l}_e(t) - l_e(t)\| < 0.45$m and $\sqrt{r^2 - h^2} = 0.4$m, a reasonable choice could be $\epsilon_l = 1$m.

IV. Security Analysis

Portunes is not designed to be secure against inside attacks. If the attacker compromises the CSP or the PO, he is able to disrupt dynamic charging of an EV or on a charging section, respectively. If the attacker compromises charging pads he may obtain the entire key set, but compromising the PO or charging pads does not threaten the EV's location privacy due to using pseudonyms. Assuming that the CSP, the PO, and the charging pads are trustworthy, in the following we discuss how *Portunes* mitigates various outside attacks.

A. Replay Attack for Electricity Theft

An attacker may capture the beacon (msg 5) sent by EV e to pad p, and replay the beacon to a pad p'. For a pad p' to validate the beacon, the attacker has to replay the beacon to a nearby pad p' with $|l_{p'} - \hat{l}_e| < \epsilon_l$ (and thus $|l_{p'} - l_p| < 2\epsilon_l$) and within $2\epsilon_t$ time. Furthermore, for pad p' to switch on, either (i) the beacon of EV e was not received by pad p' due to noise or jamming (the attacker follows EV e), or (ii) EV e has not yet reached pad p' (the attacker is in front of EV e).

In case (i) the attacker has to wait for EV e to leave pad p' and should drive above pad p' itself in order to receive free charging. Assuming that EV e is 5 meters long and denoting the speed of EV e (and of the attacker) by v_e, the attacker has to be within $v_e\epsilon_t - 5 + 2\epsilon_l$ distance of EV e. At a speed of $v = 108$km/h and $\epsilon_t = 200$ms this corresponds to about 6m, which is infeasible. In case (ii) the attacker has to be in front of EV e, but within $2\epsilon_l - 5$ distance, which again is infeasible. Recall that if $|l_{p'} - \hat{l}_e| > \epsilon_l$ then pad p' does not activate, but sends msg 6 in response, which the attacker cannot deccrypt without the key $K_{f(\pi)}$.

[2]Federal standards (e-CFR 393.82) in the US allow a maximum speedometer error of 8 km/h at speed 80 km/h. If the EV broadcasts the beacon every 5 ms, i.e., $t' - t = 5$ ms, the location error introduced by speedometer inaccuracy is at most (8 km/h·5 ms =) 0.01 meter.

B. Replay Attack for DoS

Although not able to charge its EV, an attacker may replay a captured beacon immediately to a nearby pad p'. This would cause pad p' to switch on before EV e arrives to it, after which p' would not validate the beacon of EV e. This attack is, however, rather costly as in order for the attacker to perform this attack to the entire charging section, the attacker must be able to capture a new beacon every $2\epsilon_t$ time.

C. Location Privacy Attack

An attacker could attempt to (i) link the pseudonyms used by the same EV at different charging sections, and infer the victim EV's route; or (ii) infer that the same victim EV has visited a charging section repeatedly. *Portunes* defends against these attacks by assigning pseudonyms randomly to EVs. The only thing an attacker can infer is that an EV with pseudonym π is moving across a charging section, since within a charging section the EV uses the same pseudonym to communicate with all charging pads. Nevertheless, this information would be of little value to the attacker, as a charging section is typically only a few kilometers long.

V. Discussion

a) Alternative Billing Models: In the conventional pay-per-use model, the EV pays the pad owner either before (pre-paid) or after (post-paid) using dynamic charging. In the post-paid scenario, the pad owner needs to authenticate billing information provided by the EV (e.g., credit card number), which may be directly related to the EV or the driver's identity, and can be a serious privacy threat if the attacker captures such information. In the pre-paid scenario, in addition to identity authentication, the pad owner also has to verify that the EV has sufficient balance in its virtual wallet, which further complicates the protocol design. Moreover, if the EV drives to another city, the pad owner there may have to learn about this EV first (e.g., obtaining its public key certificate from a Certificate Authority) before being able to authenticate its identity, which incurs additional communication overhead.

b) Trustfulness: In *Portunes* the EV pays for the amount of energy reported by the PO. To discourage the PO from overclaiming, the CSPs could levy a fine on a PO that is caught overclaiming, e.g., when test-driving through the PO's charging section.

An alternative solution would be that both the EV and the PO report the amount of energy charged to the CSP, possibly in real time. The CSP could compare the reports and detect if the pad owner overclaims or the EV underclaims. This approach incurs more communication (EV to CSP), and over/underclaim detection is complicated by the fact that energy loss during dynamic contactless charging inherently causes the PO and the EV to have different readings of the total energy transfer.

c) Alternative Defense against Replay Attack: A key element in *Portunes* is replay attack mitigation through location and time information in the beacons. An alternative design could be that whenever a charging pad p successfully authenticates an EV, it would broadcast pseudonym π and the EV's speed to all the other charging pads, which could then estimate the EV's future location. Assuming a beacon is valid within ϵ_t time after it is sent, if the attacker captures a beacon at pad p, it would have to replay the beacon to a pad q that the EV could reasonably reach in ϵ_t time.

This alternative design would not need the EVs to include their location estimates in the beacons. It does, however, require that a pad sends a broadcast message whenever it authenticates an EV, and thus the number of messages sent would scale linearly with the number of EVs and with the number of pads. One could reduce the number of broadcasts (e.g., by requiring only every 1 out of 10 pads to broadcast), and the set of recipient pads through imposing a speed limit. The major drawback of this alternative design compared to *Portunes* is that it requires pad-to-pad communication, which exposes a new attack vector as the attacker could attempt to forge pad-to-pad control messages. Inter-pad control messages would thus have to be authenticated, which requires more computational power in the pads and may also complicate the protocol design.

VI. Evaluation

We implemented *Portunes* on RaspberryPi Model B [13] using Crypto++ 5.6.2 library. The RaspberryPi features 700 MHz CPU and 512 MB RAM, and costs less than $40 (USD). In Fig. 2 we compare the generation and verification time of the beacon (msg 5) using *Portunes* and Elliptic Curve Digital Signature Algorithm (ECDSA) respectively. We use AES with CFB mode and 128-bit key as the symmetric encryption algorithm in *Portunes*, and use ECDSA on P-224 curve, which results in 448-bit signature. Both *Portunes* and ECDSA provide 112-bit security strength in this setup.

Using *Portunes*, it takes 0.34 ms to generate a 100-byte beacon, and 0.13 ms to verify the beacon. The generation and verification time increase to 0.48 ms and 0.25 ms respectively as the beacon size increases from 100 to 900 bytes. In practice the beacon size would depend on the semantic parameters contained in the req field.

In comparison, for all beacon sizes, ECDSA takes over 25 ms to generate a signature, and over 40 ms to verify a signature. This implies that using ECDSA, the EV can only broadcast a beacon every 25 ms, and the charging pad must spend another 40 ms to verify the signature before it starts charging the EV. This makes ECDSA infeasible in our scenario, where the total communication contact time between the EV and the pad may be only tens of milliseconds.

Since an EV must authenticate with the charging pad before it can be charged, we finally consider the authentication success probability under different EV speeds and beacon broadcast frequencies. Let s denote the packet drop probability, b denote the EV's beacon broadcast frequency, and T denote the time that the EV stays within communication range of the pad. Then the EV broadcasts a total of $N = bT$ beacons within T. Assuming independent packet drops, we can express the authentication success probability τ as

$$\tau = 1 - s^N = 1 - s^{bT} = 1 - s^{b \cdot \frac{2\sqrt{r^2 - h^2}}{v}} \qquad (12)$$

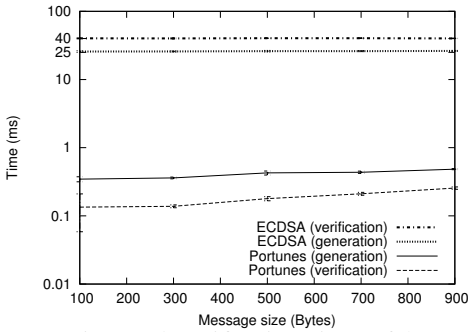

Fig. 2: Generation and verification time of beacon (msg 5) using Portunes and ECDSA respectively. Error bars indicate 95% confidence intervals.

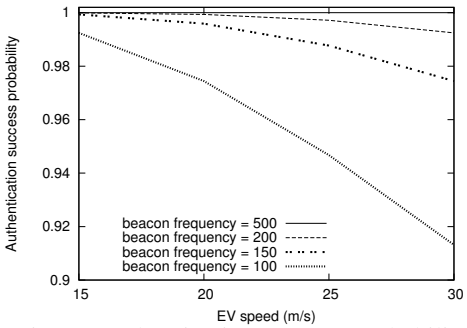

Fig. 3: Authentication success probability.

where r is the communication range of the wireless devices, h is the vertical distance from the ground to the EV's wireless device, and v is the EV's speed. In Fig. 3 we plot the authentication success probability vs. the EV speed for various beacon broadcast frequencies for $r = 0.5$ m, $h = 0.3$ m, and $s = 0.4$. The figure confirms that the success probability decreases exponentially with increasing speed, but it also shows that the success probability can be kept constant by increasing the beacon frequency linearly with the speed.

VII. Related Work

Key pre-distribution based authentication was primarily used in wireless sensor networks [14], and has also been adapted to vehicular network [15]. Our work differs in that EV e is authenticated using a single key $K_{f(\pi)}$ instead of a subset of keys, and incurs less overhead during key transmission. One-time signatures [7], [8] only allow the EV to sign one or several messages using the same key. In our scenario this would imply that a single EV needs thousands of keys in order to authenticate with each charging pad in the charging section, which incurs considerable key generation and distribution cost and is impractical. FastAuth [6] limits the message content to vehicle's location and speed, whereas *Portunes* allows the EV to include arbitrary information, such as battery type and desired charging rate, in the beacon (msg 5). HIP-based solutions [16], [17] for micro-mobility would incur non-trivial overhead during authentication handover between charging pads, and are infeasible in our scenario where an EV encounters a new pad every tens of milliseconds. RSU-based privacy-preserving authentication [18], [19] for VANET generally requires the vehicle to negotiate with an RSU to obtain a temporary session key. This is similar to our case where the CSP allocates pseudonym π and key $K_{f(\pi)}$ to an EV before it enters a charging section. *Portunes* differs from existing works in that the key $K_{f(\pi)}$ is pre-distributed to all the charging pads before it is assigned to an EV. This provides seamless authentication handover, which is crucial in dynamic charging where an EV must authenticate with a new charging pad every tens of milliseconds.

VIII. Conclusion

In this paper we proposed *Portunes*, an authentication protocol for dynamic contactless EV charging. *Portunes* preserves the EVs' location privacy and supports roaming for the charging service. It allows the EVs to perform lightweight authentication with the charging pads and needs no pad-to-pad communication as it verifies the spatio-temporal location of the EVs. The implementation on Raspberry Pi indicates that message generation and verification using *Portunes* are both over 80 times faster than using ECDSA. Our security analysis shows that *Portunes* effectively mitigates outside attacks, and numerical results show that *Portunes* is both computationally efficient and can enable reliable charging.

IX. Acknowledgment

We thank Prof. Philip T. Krein for providing the dynamic charging scenario and for his insightful comments.

References

[1] H. Wu, A. Gilchrist, K. Sealy, P. Israelsen, and J. Muhs, "A review on inductive charging for electric vehicles," in *IEMDC*, May 2011.
[2] H. Perik, "Practical EV Integration Cases for Static and Dynamic Wireless Power Transfer," in *IETEVC*, 2013.
[3] G. Covic and J. Boys, "Modern Trends in Inductive Power Transfer for Transportation Applications," *IEEE ESTPE*, 2013.
[4] S. Lee, J. Huh, C. Park, N.-S. Choi, G.-H. Cho, and C.-T. Rim, "On-Line Electric Vehicle using inductive power transfer system," in *ECCE'10*.
[5] S. Ahn and J. Kim, "Magnetic field design for high efficient and low EMF wireless power transfer in on-line electric vehicle," in *EUCAP '01*.
[6] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Flooding-resilient broadcast authentication for VANETs," in *ACM MobiCom*, 2011.
[7] L. Reyzin and N. Reyzin, "Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying," in *ACISP*, 2002.
[8] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *IEEE SP*, 2000.
[9] "OCRA: OATH Challenge-Response Algorithm," *RFC 6287*.
[10] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *MobiCom*, 2010.
[11] E. Mallada, X. Meng, M. Hack, L. Zhang, and A. Tang, "Skewless network clock synchronization," in *IEEE ICNP*, 2013.
[12] "Global Positioning System Standard Position Service (SPS) Performance Standard, 3rd edition," 2008.
[13] "RaspberryPi." [Online]. Available: www.raspberrypi.org
[14] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," in *IEEE SP*, 2003.
[15] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing privacy using symmetric random key-set in vehicular networks," in *ISADS*, 2007.
[16] "RFC 5201: Host Identity Protocol."
[17] Z. Gurkas Aydin, H. Chaouchi, and A. H. Zaim, "eHIP: early update for Host Identity Protocol," in *Mobility '09*.
[18] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs," in *SECON '09*.
[19] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications," in *IEEE INFOCOM '08*.