PRIVACY-PRESERVING AUTHENTICATION AND BILLING FOR DYNAMIC CHARGING OF ELECTRIC VEHICLES

BY

HONGYANG LI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Professor Klara Nahrstedt, Chair
Professor Carl A. Gunter
Associate Professor Nikita Borisov
Associate Professor György Dán, KTH Royal Institute of Technology

ABSTRACT

Dynamic charging of electric vehicles (EVs) is a promising technology for future electrified transportation. By installing wireless charging pads under the roadbed, dynamic charging allows EVs to charge their batteries while moving through magnetic induction between the wireless charging pad and the receiving coil attached to the EV's battery. A pre-requisite for dynamic charging in practice is the support of cyber infrastructure and protocols. Although many research efforts aim to increase the charging efficiency and remove the physical barriers of dynamic charging, protocols in the cyber space that support dynamic charging is still lacking, especially protocols for digital authentication and billing. Due to EV's high mobility, location privacy is also an important research issue. In this thesis we present three protocols: FADEC, Portunes, and Janus, that together provide privacy-preserving authentication and billing framework for dynamic charging of EVs. The protocols are tailored towards the dynamic charging scenario to reduce real-time computation and communication overhead, and uses modern cryptography building blocks to preserve the EV's location privacy. Simulation results and implementations indicate that the presented protocols are efficient and feasible for future dynamic charging applications.

To my wife, Siting.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Electric vehicles (EVs) have many benefits compared to conventional combustion engine vehicles: they are very quiet, offer high torque, and most notably they produce no tailpipe emissions. The major disadvantage of EVs today is their limited range and their longer battery charging time compared to conventional vehicles. The EV's expensive battery constitutes a large part of its price and makes it less competitive in the market. In these aspects, recent advances in *dynamic charging* technology, which allows EVs to charge their batteries while moving on the road, helps address some of the major drawbacks of EVs.

Dynamic charging technology charges the EV's battery through magnetic induction: a charging section is a road segment with a sequence of wireless charging pads installed under the roadbed, and as the EV moves along the charging section, the magnetic induction between the roadbed charging pad and the coils, attached to the EV's battery, charges the EV's battery. By allowing EVs to charge while moving, dynamic charging alleviates the problem of short driving range of today's EVs. With enough coverage of charging sections, dynamic charging also reduces the required battery size of an EV and in turn reduces its price, which makes EVs more affordable to customers.

Dynamic charging has attracted attention from both the industry [1] and the research community [2, 3, 4]. Several research efforts have been going on to bring dynamic charging to practice: Oak Ridge National Lab (ORNL) has demonstrated 6.6kW dynamic charging with 85% efficiency over 16 cm air gap, and is currently integrating the dynamic charging technology into Toyota RAV4 SUV [5]. The Online Electric Vehicle (OLEV) project in the Korean Advanced Institute of Science and Technology (KAIST) developed Shaped Magnetic Field in Resonance (SMFIR) Technology that delivers power wirelessly from roadbed charging pads to the battery of electric buses. In 2013 two OLEV buses were deployed along a 15-mile inner-city route in Gumi, Korea. The buses receive up to 100 kW power at 85% transfer

Figure 1.1: Illustration of dynamic charging of electric vehicles [8]. The EVs charge their batteries by moving along the left-most lane.

efficiency with a 20 cm air gap between the bus and the road surface [6]. The UK has also started testing dynamic charging for electric vehicles [7]. In Figure 1.1, we illustrate the concept of dynamic charging of an electric vehicle.

Despite its many advantages, dynamic charging comes with its own limitations. At the current stage, the charging efficiency is sensitive to many physical parameters including the EV's position, its movement speed, and the air gap between the EV and the charging pad. The EV must be properly aligned with the charging pad. The current prototypes of dynamic charging require the EV to move at constant speed (e.g., 30 km/h) in order to achieve the desired charging efficiency. The maximum air gap supported by most dynamic charging systems is around 15-20 cm. In the future, the dynamic charging section is to be used by different types of EVs that have different sizes, different air gaps between the EV's receiving coil and the charging pad, and move at different speeds. The dynamic charging system must learn the EV's parameters such as its speed and airgap, switch on each individual charging pad just before the EV comes, adjust the power output according to the incoming EV's parameters, and switch off the charging pad after the EV moves over. However, before the EV can send its parameters to the charging pads, the EV and the charging pads must properly authenticate each other. Since the EV's parameters may contain sensitive information such as its current battery State-of-Charge (SoC), which can

be used to infer the EV's past trajectory, the EV must make sure that the other communicating party is indeed a valid charging pad before disclosing its charging parameters. Similarly, the charging pad must authenticate the EV before switching on and charging; otherwise a malicious attacker may send forged messages to charging pads and cause them to switch on while there is no EV above them, which in turn causes energy waste and safety concerns. Although many research efforts have been going on to improve the charging efficiency and to loosen the restrictions of physical parameters of dynamic charging, digital authentication is a less researched area for dynamic charging of electric vehicles.

Billing is another important issue that is overlooked in the current research of dynamic charging. In certain scenarios such as the Online Electric Vehicle developed by KAIST, the dynamic charging is used only by electric buses, and both the electric buses and the dynamic charging infrastructure are operated by the same entity. However, future dynamic charging aims at serving individual EVs, in which case correctly billing the individual EV drivers for their use of dynamic charging service remains a challenge. Due to the nature of dynamic charging that allows EV to charge its battery while moving, billing for dynamic charging is more difficult than billing for static charging. Today's static charging service usually adopts a pay-per-use billing model: the EV stops at the charging station to charge its battery, at which time the driver could pay the charging station by cash or credit card. In this thesis, we envision a subscription-based billing model for dynamic charging that draws inspiration from today's billing model for cellular service. Similar to the cellular service billing model, where a mobile phone user makes or receives calls at multiple location within cellular signal coverage and pays a single monthly bill, the EV receives dynamic charging service from various charging pad owners at different times and locations, and a third party such as the utility could aggregate the EV's charging activities in a monthly bill. The EV will pay the utility for the dynamic charging service it received during the past month, and the utility in turn pays each charging pad owner accordingly. The subscription-based billing model allows the utility to treat the EV and other appliances in a uniform way, and also facilitates implementation of flexible pricing options, e.g., the utility could apply discounts to the EV's dynamic charging bill if the EV has enrolled in the vehicle-to-grid (V2G) program that helps the utility to reduce peak load.

What makes authentication and billing for dynamic charging more challenging is the concern for EV's location privacy. In general, a privacy-

preserving design should not allow an outside observer to learn or infer useful information about the EV's locations. One example that violates the EV's location privacy is the use of EV's long-term public key for authentication: if the EV always uses the same public key to authenticate with other charging-related entities at different locations, any outside observer can eavesdrop on the wireless communication channel and learn the trajectory of the EV by tracking the use of the same public key. The billing protocol must also preserve the EV's location privacy. In particular, the billing protocol should allow the utility to calculate each individual EV's total monthly bill without learning when and where the EV has been.

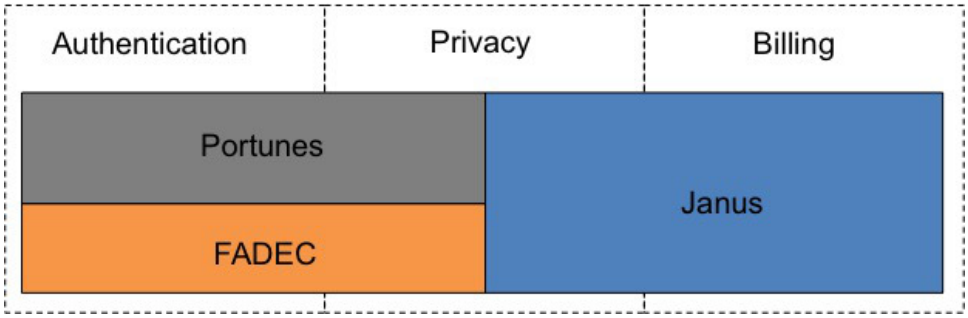

Figure 1.2: Illustration of the problems that FADEC, Portunes, and Janus aim to solve respectively.

In this thesis we present our effort towards privacy-preserving authentication and billing for dynamic charging of electric vehicles. Our contribution consists of three protocols: FADEC, Portunes, and Janus. In Figure 1.2 we illustrate the problem space of each protocol. Below we briefly describe each protocol. A more elaborated overview of our contributions can be found in Chapter 3.

- **FADEC** is an efficient real-time authentication protocol that enables EVs to authenticate with the utility through roadside units (RSUs). FADEC uses lightweight symmetric cryptographic operations to enable efficient authentication between EVs and RSUs and between EVs and the utility, and adopts a proactive key dissemination approach to achieve seamless handoff authentication between the same EV and different RSUs.

- **Portunes** is an efficient real-time privacy-preserving authentication protocol that provides mutual authentication between EVs and charging pads. Portunes adopts a key pre-distribution approach to minimize the computational cost of signature generation and verification during the authentication between EV and charging pads.

- **Janus** is a privacy-preserving billing protocol that allows utility to correctly calculate and bill the EV without learning the EV's trajectories. Janus uses modern cryptographic tools such as homomorphic encryption and blind signatures with attributes to ensure that the EV's total bill is calculated correctly without revealing when and where the EV has charged its battery.

The rest of the thesis is organized as follows: in Chapter 2, we briefly introduce the dynamic charging technology; in Chapter 3, we give a brief overview of our contributions; in Chapter 4, we describe the FADEC protocol that provides fast authentication between EV and the utility; in Chapter 5, we describe the Portunes protocol that achieves efficient and privacy-preserving authentication between EV and charging pads; in Chapter 6, we describe the Janus protocol that enables privacy-preserving billing without revealing the EV's locations to the utility; we include discussions and related works in the corresponding chapter of each protocol, and conclude the thesis in Chapter 7.

CHAPTER 2

DYNAMIC CHARGING

2.1 Overview of Electric Vehicle Charging

Depending on how the EV is connected to the power grid, we can classify EV charging as wired charging and wireless charging. Depending on whether the EV is stationary or moving while charging, we can classify EV charging as static charging and dynamic charging.

Static wired charging is the most widely used method of EV charging at the moment. According to the Society of Automotive Engineers (SAE) standards [9], charging can be classified as AC charging and DC charging. AC charging is slower and normally used for charging at residential places, and can be further divided into AC Level 1 and AC Level 2 charging, where AC Level 1 provides charging rate of 5 miles per hour (i.e., charging the EV continuously for an hour will give the EV enough electricity to drive 5 miles) and AC Level 2 provides charging rate up to 60 miles per hour. DC charging is usually used at fast charging station. DC Level 1 charging can charge the battery up to 120 miles per hour, and DC Level 2 charging can charge up to 300 miles per hour. In Table 2.1, we summarize the charging categories according to the SAE standards.

Wireless charging allows the battery to be charged without attaching any cable to the power source. Wireless charging is based on the following laws of physics: (i) a closed circuit loop carrying a current generates a magnetic field around the loop; and (ii) a coil intersecting a magnetic field generates a voltage in the coil. The idea of using magnetic coupling to transfer electric-

Charging Level	Setting	Charging Rate
AC Level 1	Residential/Parking Lot	5 miles / hour
AC Level 2	Residential/Commercial	10 - 60 miles / hour
DC Level 1	Commercial	120 miles / hour
DC Level 2	Commercial	300 miles / hour

Table 2.1: Charging levels according to SAE standards [9]

Figure 2.1: Plugless wireless charging system for Nissan Leaf showing the wireless charging pad and the wall-mount control panel [10].

ity over an airgap was first introduced by Nikola Tesla about a century ago, and is called inductive power transfer nowadays. Many small appliances such as smart phone, smart watch, electronic toothbrush, etc. have already adopted wireless charging, and static wireless charging for electric vehicles are slowly becoming available to customers. For example, Plugless [10] provides wireless charging system for certain models of EVs including Nissan Leaf, Chevrolet Volt, and Cadillac ELR, for a price between 1200 USD and 1900 USD depending on the EV model. In Figure 2.1, we show a picture illustrating the Plugless wireless charging system for Nissan Leaf. The installation consists of a wall-mount control panel, a wireless charging pad on the ground, and an adapter inside the vehicle.

2.2 Dynamic Charging

Dynamic wireless charging, or simply dynamic charging, takes static wireless charging one step forward. It shares the same basic principle of inductive power transfer as static wireless charging, but the wireless charging pads are

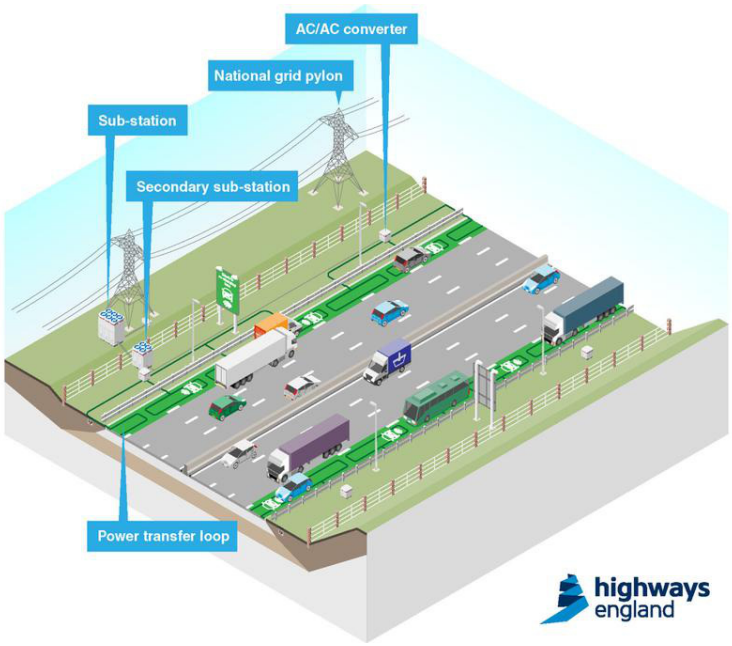

Figure 2.2: Illustration of dynamic charging infrastructure that UK is planning on testing [11].

installed under the roadbed covering a road segment of several kilometers, and the EV can charge its battery wirelessly by moving over the charging pads. In Figure 2.2, we illustrate a sample infrastructure model for dynamic charging that UK is planning on testing [7].

We define a *dynamic charging section* to be a straight road segment under which the wireless charging pads are installed. We assume a dynamic charging section has a single lane with only one entry and one exit. To facilitate power management, each charging section is typically several kilometers long. Within the charging section, short wireless charging pads (e.g., 40 cm long) are placed consecutively under the roadbed, with tens of centimeters between each other. In addition, each charging pad can be individually switched on and off, independent of other charging pads. Ideally, the charging pad should switch on just before the EV moves above it, and should switch off immediately after the EV moves away.

We assume the dynamic charging sections are operated by *Pad Owners* (POs). The PO may either produce electricity, or may purchase electricity from the utility. We assume that each dynamic charging section is owned by exactly one PO, but a PO can own multiple dynamic charging sections, and there can exist multiple POs operating different dynamic charging sections in the same area.

We define a *dynamic charging session* to be the continuous time period

starting from the moment that EV e starts charging its battery from a dynamic charging section operated by PO p, until the moment that the EV stops charging from the same charging section. During a single dynamic charging session, the EV continuously charges its battery by moving over a series of wireless charging pads in the dynamic charging section.

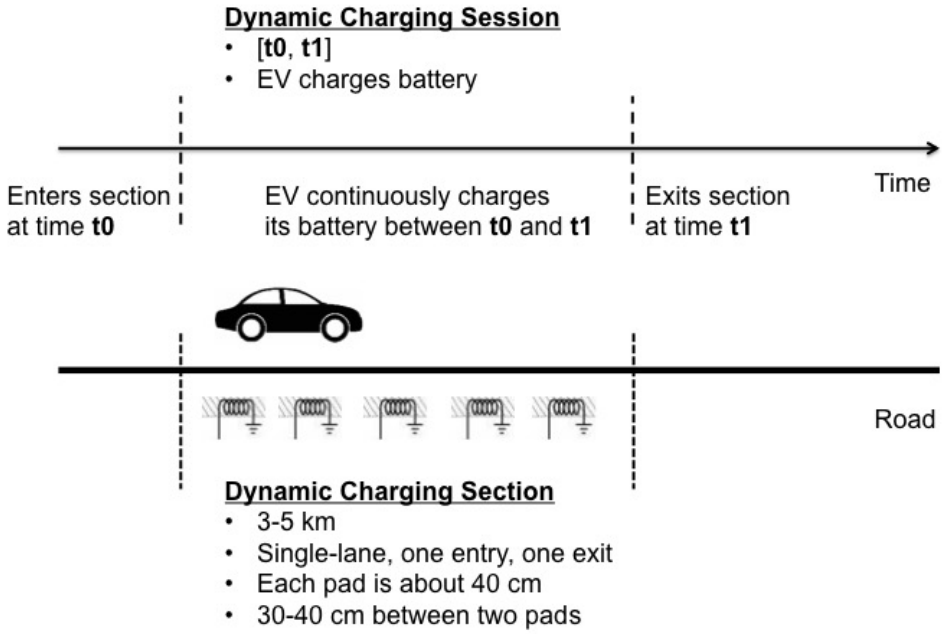

Figure 2.3: Illustration of Dynamic Charging Section and Dynamic Charging Session.

In Figure 2.3 we illustrate the definition of dynamic charging section and dynamic charging session.

2.3 Advantages, Limitations and Challenges of Dynamic Charging

Dynamic charging has many advantages compared to other modes of charging. Today most EVs are used for short-range inner-city commutes, and for longer distance trips the driver must carefully plan where to stop and charge the EV to avoid running out of battery in the middle of the trip. The ability to charge EVs while moving greatly reduces the driver's anxiety about the EV's driving range as well as reduces the trip planning effort. Since the EV can charge its battery on the road, dynamic charging also reduces the EV's battery size that is necessary for daily use. Since the battery constitutes a large portion of the EV's total cost, a reduced battery size would in turn

reduce the cost of EV and make it more affordable and competitive in the vehicle market.

However, dynamic charging comes with several limitations. Today's most advanced dynamic charging system can only achieve about 70-80% charging efficiency of wired charging. The efficiency of dynamic charging is also affected by many physical and environmental conditions, including the alignment of the EV with the charging pad, the size of the airgap (i.e., the distance between the receiving coil attached to the EV's battery and the roadbed wireless charging pad), and the speed of the EV. The EV must be properly aligned with the charging pad in order for dynamic charging to happen. This means that dynamic charging is likely to lose efficiency whenever the EV is not moving along a straight lane, e.g., changing lanes, or when the EV is not moving at constant speed. Compared to other modes of charging, dynamic charging requires large investment of infrastructure since the wireless charging pads need to be installed under the roadbed.

Dynamic charging also brings various unique challenges to the cyber infrastructure. A dynamic charging system intended to serve general EVs must be able to charge different types of EVs with different battery types, different sizes of airgap, desired voltage, etc. This requires proper digital communication support so that the dynamic charging system can learn the necessary parameters of the incoming EV. The dynamic charging system must be able to properly identify and authenticate the EVs for billing purpose, which is challenging due to EV's high mobility and the requirement to preserve EV's location privacy.

We observe that the above challenges represent a knowledge gap between a feasible dynamic charging system for general EVs and the current research effort in dynamic charging. Today's research effort of dynamic charging mostly focuses on increasing the charging efficiency and removing/reducing the limitations of dynamic charging (e.g., increasing the maximum speed or the airgap allowed), while the challenges in the cyber space are mostly ignored. For example, the OLEV system deployed in Korea ignores the variation of general EVs that may come with different battery types and other physical parameters, and only focuses on special electric buses. In the OLEV system, the charging infrastructure is deployed and operated by the same entity that operates the electric buses, which means that no external billing is required between the charging facility and the EVs. Since no vehicles other than the OLEV electric buses are able to use the dynamic charging system, digital authentication is not necessary either, as no other vehicle can steal energy from the dynamic charging system. A future dynamic charging

system intended for the general public would violate all the above assumptions: the system must consider the variation of EV's charging parameters such as desired voltage and airgap, and must be able to distinguish between EVs and non-EVs as well as authenticate EVs and bill the correct customer.

2.4 Subscription-based Billing Model for Dynamic Charging

Billing model is another important research issue for dynamic charging, and impacts the designs of both the authentication and the billing protocols. In this section, we describe a subscription-based billing model similar to today's cellular service, where the EV pays a single bill once every month rather than making payments for each dynamic charging session individually. Our model involves three types of entities: the utility, the EV, and the pad owners. The pad owners are the ones that own and operate the dynamic charging infrastructure. There can be many pad owners in the same region.

The billing model consists of two operations: fee negotiation and fee aggregation. Fee negotiation happens prior to each dynamic charging session, where the EV and the PO negotiate and agree on the charging fee that the EV should pay for the coming dynamic charging session. Fee aggregation happens only once at the end of each billing cycle, where the EV calculates and submits to the utility its total fee that it should pay to the utility, and the PO calculates and submits to the utility the total fee that it should receive from the utility.

We illustrate our proposed billing model in Figure 2.4. EV 1 receives dynamic charging only once from pad owner A, and the charging fee for that charging session is \$3. EV 2 is involved in one dynamic charging sessions with PO A for \$4, and another charging session with PO B for \$5. From the utility's perspective, the total bill for EV 1 would be \$3, and the total bill for EV 2 would be \$9 (= 4 + 5). Since PO A provided dynamic charging to both EV 1 and EV 2, the total fee that the utility should pay to PO A is \$7 (= 3 + 4). PO B only provided dynamic charging to EV 2, and receives \$5 from the utility.

The advantage of the subscription-based billing model above is threefold: (i) it allows the utility to have a holistic view of all the EV's charging activities, including charging at home, at parking lots, at commercial charging station, and dynamic charging on the road; (ii) it allows the utility to treat the EV as part of the user's home appliances; and (iii) it enables flexible

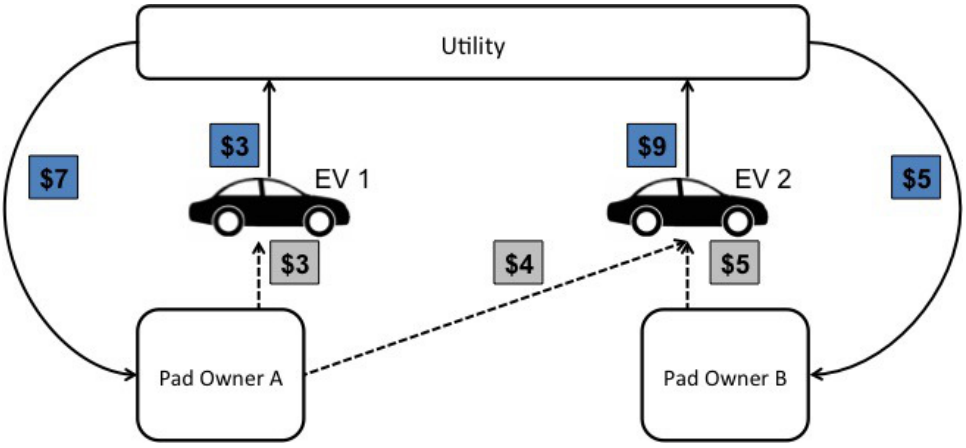

Figure 2.4: Illustration of subscription-based billing model for dynamic charging. The fees in light-colored boxes in the bottom row ($3, $4, $5) are the result of fee negotiation. The fees in dark-colored boxes in the top row ($7, $3, $9, $5) are the result of fee aggregation.

pricing plan for the EVs. Having a holistic view of the EV's charging activities allows the utility to better understand the charging demand and reduce peak load, while the ability to treat EV as a special home appliance enables flexibility in pricing plans. For instance, if the user chooses to join the vehicle-to-grid (V2G) program that helps the utility to reduce peak load, the utility could apply discounts to the EV's dynamic charging bill. The subscription-based billing model also enables flexible pricing plan similar to the data plan model in today's cellular service. For example, the EV could purchase a plan of 1000 miles from the utility, and the EV can use dynamic charging anywhere anytime to recharge its battery up to 1000 miles of total driving distance.

CHAPTER 3

OVERVIEW AND CONTRIBUTIONS

In the previous chapters we have introduced the dynamic charging technology, discussed its advantages and limitations, and the necessary support of cyber infrastructure and protocols. The thesis thus focuses on the following statement:

> Dynamic charging is the next-generation cyber-physical technology for electrified transportation that requires new designs of authentication and billing protocols to enable secure and privacy-preserving communication and billing.

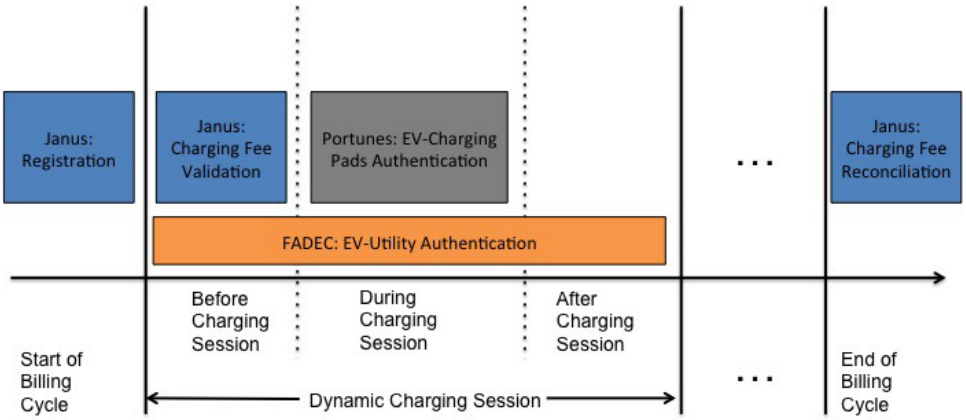

Figure 3.1: Framework Overview. The billing cycle contains one starting period, one ending period, and many dynamic charging sessions in between. The starting period is used for preparation work, e.g., the utility issue anonymous credentials to the EVs that will be used later. During each dynamic charging session, FADEC takes care of EV-Utility authentication, Portunes takes care of EV-Charging Pad authentication, and Janus generates cryptographic receipts for the charging fee. During the ending period, Janus reconciles the total charging fees.

The contribution of this thesis consists in three major protocols: FADEC, Portunes, and Janus. FADEC is a general V2I authentication protocol that aims to provide seamless authentication between EV and a series of roadside

units (RSUs). Portunes is a privacy-preserving authentication protocol that allows EVs to authenticate with charging pads without revealing the EV's true identities to the charging pads. Janus is a privacy-preserving billing protocol that allows the utility to calculate the EV's monthly bill without learning the EV's locations. In Figure 3.1, we give an overview of the framework and illustrate which part of the protocols is executed during different time of the billing cycle. We give a brief overview of each protocol below.

- **FADEC** is a real-time authentication protocol that aims to provide efficient authentication between EVs on the road and the utility. The real-time authentication between EV and utility in turn enables secure real-time communication, which can be useful for a variety of scenarios, e.g., the EV can upload its battery statistics to the utility for real-time diagnosis, and the utility can broadcast dynamic pricing information to EVs on the road. FADEC assumes a setting where the EVs communicate to the utility with the help of roadside units (RSUs), which relay messages between the EV and the utility. The major challenge that FADEC solves is the authentication handoff between RSUs when the EV exits the communication range of one RSU and enters that of another. FADEC adopts a proactive key dissemination approach that provides seamless authentication handoff, thus reducing the need to renegotiate session keys between EV and RSUs. FADEC can be viewed as a general Vehicle-to-Infrastructure (V2I) authentication protocol, and can be extended to various scenarios other than dynamic charging.

- **Portunes** is a real-time authentication protocol designed specifically for dynamic charging, and provides efficient authentication between EVs and wireless charging pads. One challenge in the dynamic charging scenario is that the EV encounters charging pads very frequently (e.g., every 30 ms), and the contact time between the EV and each charging pad is short. In order to complete authentication within the short contact time, the authentication protocol must use lightweight cryptographic operations in real time. Portunes achieves this by adopting a key pre-distribution approach, where the computationally intensive operations such as key generation are performed during the night when most vehicles are parked, and pre-distributes authentication materials to the charging pads, which reduces the effort of real-time key negotiation between EV and charging pads and thus achieves fast lightweight real-time authentication.

- **Janus** is a privacy-preserving billing protocol that provides a way for the utility to calculate the EV's monthly bill without learning its whereabouts, thus preserving the driver's location privacy. Janus uses modern cryptographic building blocks to construct homomorphic payment tokens which the EV can use to prove to the utility that the total sum of the bill is calculated correctly. By using single-use anonymous credentials, Janus also allows the utility to detect if an EV is intentionally omitting one or multiple payment in the calculation of the total bill. While there exist privacy-preserving billing protocols for other transportation scenarios such as electronic toll pricing, public transportation (trains, buses, etc.), and static charging of EVs, to the best of our knowledge, Janus is the first billing protocol proposed for dynamic charging.

CHAPTER 4

FADEC

There are many situations when the EV wants to communicate with the utility during dynamic charging: the EV could report its battery usage and the utility could use the reports to monitor the health of the charging pads and to optimize their efficiency by setting parameters, such as pulse signals and resonant frequency, in real time; the utility could also detect energy theft by checking whether the collected reports sum up to the amount of energy delivered. A natural candidate for EV to utility communication is the Dedicated Short Range Communication (DSRC), which is a medium range wireless technology developed for automotive use based on the IEEE 802.11p standard. In DSRC, roadside units (RSU) are deployed along the road, and are connected to a private or public backbone network, which allows them to communicate with the utility, e.g., through the Internet. Each EV is equipped with an on-board unit, which it uses to communicate with the RSUs, typically within a range of around 500 meters. Clearly, EVs would have to authenticate with the RSUs to ensure they send their reports to the right RSU (instead of to an attacker impersonating an RSU). At the same time, the RSUs would have to authenticate messages received from the EVs to be able to implement access control. Signing messages and verifying signatures must be fast, since the RSUs would have to handle the authentication of reports from many EVs. The authentication mechanism also needs to support mobility, because an EV could communicate with the utility through different RSUs as it moves along a road. The EV and the utility must also mutually authenticate each other. The EV must make sure that the other communicating entity is indeed the utility before sending messages that may contain sensitive information such as its battery State-of-Charge (SoC), and the utility must also authenticate the EV to tell legitimate messages from fake messages generated by a malicious attacker.

The IEEE 802.11p standard suggests the use of Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication in vehicular networks. Recent work [12] has shown, however, that using ECDSA it could take a sig-

nificant amount of time to sign a message and to verify a signature, which makes it susceptible to DoS attacks. To get around the computational overhead of ECDSA, recent works proposed the use of one-time signature for authentication [13, 14, 12]. However, one-time signature is not the ideal solution in our scenario since it could incur non-trivial key generation and signing overhead [13], requires delayed verification [14], or puts restrictions on the content to be authenticated [12].

In this chapter, we describe *Fast Authentication for Dynamic EV Charging (FADEC)* designed to support the communication between the EV and the utility during dynamic charging. FADEC features fast message signing, fast signature verification, fast hand-off authentication, and low communication overhead. FADEC allows the EV to use the same key to authenticate with a series of RSUs, so that the EV does not re-authenticate itself every time it encounters a new RSU, without sacrificing security. Our simulations show that FADEC is suitable for dynamic EV charging scenarios. Compared with ECDSA, FADEC reduces the data delivery delay by up to 97% and improves the delivery ratio by more than an order of magnitude.

The rest of this chapter is organized as follows. In Section 4.1, we introduce security background; in Section 4.2, we describe our system model and assumptions; in Section 4.3, we describe the proposed authentication solution; in Section 4.5, we present simulation results; in Section 4.7, we review related work; and finally we conclude this chapter in Section 4.8.

4.1 Security Background

4.1.1 HMAC

Hash-based Message Authentication Code (HMAC) is an authentication scheme that relies on a symmetric key k shared between the sender and the receiver. When the sender wants to send a message M, he computes a hash value $HMAC(k, M)$ using the shared key k on the message M. Both M and $HMAC(k, M)$ are sent to the receiver. Upon receiving message M' and its signature $HMAC(k, M)$, the receiver can verify that $M' = M$, and the message comes from the authentic sender, by recomputing $HMAC(k, M')$ and verifying that $HMAC(k, M') = HMAC(k, M)$. HMAC authentication is fast, compared to public key-based authentication, and is able to achieve 112-bit security strength with proper selection of keys and hash functions [15].

4.1.2 ECDSA

In Digital Signature Algorithm (DSA), each communication party has a public key P and a private key S. The public key is made known to everyone while the private key should be known only to the owner. The sender signs the message M using his private key S to produce a signature $S(M)$, and sends it with message M. The receiver, when receiving $M', S(M)$, could verify the authenticity of the message by computing $P(S(M))$ using the public key P of the claimed sender and can verify that $M' = P(S(M))$.

Elliptic Curve Digital Signature Algorithm (ECDSA) is a DSA based on elliptic curve cryptography. The IEEE 802.11p standard suggests the use of ECDSA to authenticate vehicle safety messages. However, previous work [12] has shown that ECDSA takes non-trivial time to sign and to verify a signature, and is not suitable when there are lots of signatures to verify, which is common in scenarios where many EVs send frequent reports. Another major drawback of ECDSA is its vulnerability to DoS attacks, where the attacker could flood the network with many fake signatures, and the recipient RSU will be busy verifying those fake signatures.

4.1.3 Diffie-Hellman Key Exchange

Diffie-Hellman key exchange (DHKE) allows two parties to establish a common secret. In its simplest form, Alice and Bob, engaging in Diffie-Hellman key exchange, first agree on a common base g. Alice generates a secret x and sends g^x to Bob. Bob generates a secret y and sends g^y to Alice. Both Alice and Bob are now able to compute the common secret $g^{xy} = (g^y)^x = (g^x)^y$. The naive implementation of Diffie-Hellman does not let Alice and Bob authenticate each other, and is vulnerable to man-in-the-middle (MitM) attack. Implicitly Authenticated DHKE (IA-DHKE) defeats MitM attacks by using digital signatures [16] or incorporating the public key of the intended communicating parties in the shared secret [17]. As a result IA-DHKE does not provide anonymity.

4.1.4 Just Fast Keying (JFK)

JFK [18] is a Diffie-Hellman based key exchange protocol. The goal of JFK is to allow two communicating parties to establish a shared secret key even when the communication media is insecure, i.e., the attacker could eavesdrop on the communication channel. Compared to the original Diffie-Hellman key

exchange protocol, JFK messages are digitally signed to prevent man-in-the-middle attacks. The major advantage of JFK is that it is DoS-resistant and protects the RSU from signature flooding attack where the attacker sends lots of signatures for the RSU to verify so that it does not have time to verify signatures from honest vehicles.

4.2 System Model

Our system consists of a wireless charging pad beneath a stretch of a road, a set of RSUs along the stretch of road, the utility that provides power to the pad, and the EVs.

4.2.1 Communication Infrastructure

We assume that each EV has a DSRC on-board unit, which it uses to communicate wirelessly with the RSUs. An EV could potentially turn off its on-board unit in an attempt to charge the battery without being billed. One way to discourage this is to place cameras at the beginning of the charging section and take pictures of the EVs. An EV that refuses to communicate to the RSUs can be identified and levied a fine. This provides an incentive for the EVs to communicate with the RSUs and with the utility.

The RSUs and the utility are connected through a backbone network. In order to communicate with the utility, the EV will send its messages wirelessly to an RSU, which will then relay the EV's messages to the utility. If the utility wants to send a message back to the EV, it will send the message to the RSU through the backbone network. The RSU will then send the message wirelessly to the EV.

We assume that the EVs, the RSUs, and the utility all have their own public/private keys for digital signature. We also assume a public/private key pair that is shared by all RSUs, which allows an EV to verify that it is indeed communicating with an RSU, although it does not know which RSU it is. We assume a Certificate Authority (CA) that certifies all public keys. In particular, an EV only needs to store the public key of the CA, and can learn the authenticity of other public keys by verifying the corresponding certificates. We assume that a secure connection has been established between neighboring RSUs and between the utility and each RSU. FADEC thus focuses on the authentication between the EVs and the RSUs, and between the EVs and the utility. We assume that all EVs and all RSUs

have similar limited computational resources to sign messages and to verify signatures, while the utility has significantly more computational resources.

4.2.2 Attack Model

We assume that the attacker is computationally bounded and cannot forge a HMAC or reverse a one-way hash. The attacker could compromise an arbitrary number of EVs and RSUs, and obtain all their secrets including the private keys and the established session keys, but cannot compromise the CA nor the utility.

4.2.3 Objective

Our primary objective with FADEC is to allow the utility to verify the integrity of messages sent by the EVs and the identity of the sender for correct billing. Sole authentication of the EVs is, however, not enough. Without further authentication, an attacker could impersonate an RSU or the utility to capture messages containing sensitive information from EVs. The attacker could also be a malicious EV trying to hide its identity or pretending to be another EV in order to evade billing.

Thus, the considered scenario also requires that the EV authenticates the identity of the utility, to ensure the real-time reports are delivered to the proper utility. Since all messages between the EV and the utility are relayed by RSUs, the EVs and the RSUs must also authenticate each other. The authentication between the EVs and the RSUs is an important security primitive for network operations such as access control, load balancing, and accounting. Without such authentication, an attacker may flood the network with junk data and evade punishment by claiming the identity of some other EV. Authentication also ensures that the RSU will relay messages from the utility office to the correct EV.

4.2.4 Design Goals

Based on the above considerations we formulate the following design goals for FADEC.

- Fast Signing and Verification: since the EV both receives information from the utility and sends reports to the utility, both message signing and signature verification must be fast. Conventional approaches that

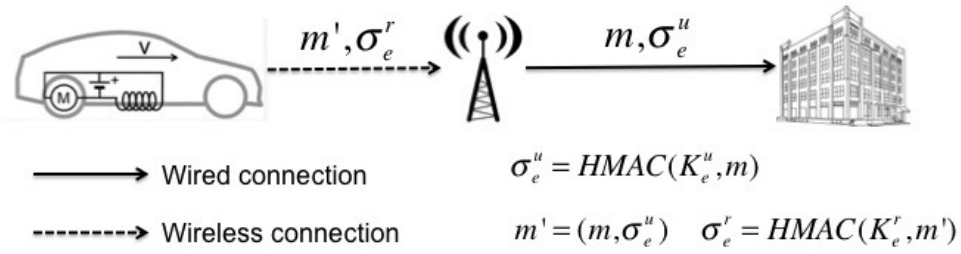

Figure 4.1: Overview of FADEC.

reduce verification overhead at the cost of increased signing effort are not suitable in our scenario.

- Fast Hand-off Authentication: when the EV is moving out of the range of the current RSU, it must be able to quickly re-authenticate itself with the next RSU so it can resume sending reports.

- Low Communication Overhead: the signature length must be short. This requirement is motivated by the condition that an EV will most likely generate many messages of small sizes, e.g., messages containing charging parameters. Attaching a long signature to a short message means high overhead and low effective spectrum utilization.

4.3 FADEC System Design

In FADEC an EV e maintains a symmetric session key K_e^r with the RSUs and another symmetric session key K_e^u with the utility. The session keys are established using JFK. Figure 4.1 illustrates the use of the keys. Before sending a message m [1] to the utility, EV e first computes the signature $\sigma_e^u = HMAC(K_e^u, m)$ on m using HMAC with key K_e^u, and the signature $\sigma_e^r = HMAC(K_e^r, m')$ on $m' = (m, \sigma_e^u)$, and sends (m', σ_e^r) to the RSU. The RSU verifies the signature σ_e^r, and then relays the message content $m' = (m, \sigma_e^u)$ to the utility through the previously established secure channel. The utility verifies the signature σ_e^u and then accepts the message m. In the following section we describe how EV e establishes the two session keys K_e^r and K_e^u.

[1] Note that FADEC does not aim to provide message confidentiality, and here m could be either encrypted or in plain text. Designing a proper encryption algorithm for dynamic EV charging is out of the scope of this chapter, although one could potentially use FADEC to establish another session key between the EV and the utility and use AES encryption.

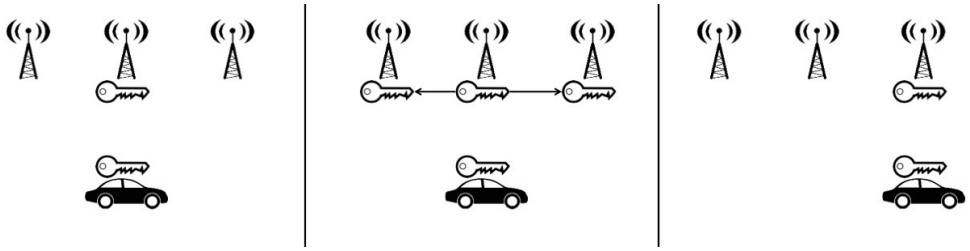

Figure 4.2: Illustration of key establishment, dissemination to neighbors and discarding of unused keys.

4.3.1 Establishing Session Key K_e^r with the RSUs

The EV establishes its session key with the RSU using JFK [18]. As the EV moves along the road, it constantly leaves the communication range of the current RSU and enters the range of a new RSU. The naïve approach would be to require the EV to establish a new session key with every RSU it encounters. However, as JFK involves digital signature computation and takes multiple rounds of message exchanges, re-establishing a new session key at every RSU would incur non-trivial computational cost to both the EV and the RSU.

To avoid key re-establishment, once the key K_e^r between EV e and the current RSU is established (using JFK), FADEC allows EV e to communicate with all the subsequent RSUs along the EV's travel path using K_e^r. FADEC achieves this by using a broadcast-and-discard approach for key dissemination, as illustrated in Figure 4.2. When RSU A first establishes key K_e^r with EV e, it broadcasts the key to all its neighbor RSUs (in terms of proximity along the road) through the backbone network. When a neighbor RSU B receives K_e^r, it stores the key for $\hat{t}_{A\to B}$ seconds, where $\hat{t}_{A\to B}$ is the *estimated time* required for an EV currently in range of RSU A to move into the range of B. If EV e does not try to communicate with RSU B using K_e^r within $\hat{t}_{A\to B}$ time then RSU B discards the key. Similarly, when C receives K_e^r, it stores the key for $\hat{t}_{A\to C}$ seconds. In Figure 4.2, EV e is moving towards C, and enters the range of C within $t_{A\to C} < \hat{t}_{A\to C}$ seconds. If EV e communicates with RSU C using K_e^r, then C will broadcast K_e^r to its neighbor RSUs, and will itself store the key for additional $\hat{t}_C$ seconds, where $\hat{t}_C$ is the *estimated time* that EV e stays within the range of C. Note that only the RSU currently associated with the EV will broadcast K_e^r to its neighbor RSUs. This prevents flooding and helps keep the RSU key storage small.

In practice, RSU B could precompute $\hat{t}_{A\to B} = \frac{d^{max}_{A\to B}}{v^{min}_{A\to B}}$, where $d^{max}_{A\to B}$ is

the maximum travel distance to enter the range of B from the range of A, and $v^{min}_{A\to B}$ is the minimum speed of an EV, if such information is available. Alternatively, the RSU may estimate $\hat{t}_{A\to B}$ based on measured times $t_{A\to B}$ to adapt to varying traffic conditions. $\hat{t}_B$ can be obtained similarly.

To estimate the number of keys stored by an RSU, observe that an RSU has a limited number of neighbor RSUs, and an RSU will disseminate only keys of associated EVs to its neighbors. In steady state, the average number of keys $\overline{N}_{A\to B}$ received by RSU B from RSU A can be expressed using Little's theorem as $\overline{N}_{A\to B} = \lambda_A \hat{t}_{A\to B}$, where λ_A is the EV arrival rate at RSU A. The EV arrival rate λ_A is bounded, and can be computed using results from traffic flow theory [19]. For example, consider that the distance between RSU A and B is $d_{A\to B}$ and the EVs travel at constant speed v_A, thus they get from RSU A to RSU B in time $t_{A\to B}$. If we denote the EV density on the road by ρ_A (EVs/mile) then the arrival rate is $\lambda_A = \rho_A v_A$ [19]. Using $\alpha_{A\to B} = \frac{\hat{t}_{A\to B}}{t_{A\to B}}$ we obtain $\hat{t}_{A\to B} = \alpha_{A\to B} d_{A\to B}/v_A$, and $\overline{N}_{A\to B} = \alpha_{A\to B}\rho_A d_{A\to B}$, which is proportional to the number of EVs between RSU A and B and to the quality $\alpha_{A\to B}$ of the estimate. Our simulations show that in a heavily loaded highway scenario an RSU needs to hold 100 - 140 keys on average. Probabilistic lower and upper bounds on the number of keys stored can be obtained using Jensen's inequality and the Edmundson-Madansky inequality, respectively, and can be used for dimensioning the RSU storage.

Compared with the mobility-prediction approach [20] for key distribution in VANET which predicts the next RSU that the EV will encounter and sends the key only to that RSU, the FADEC approach has two major advantages. First, FADEC does not need to predict the individual mobility of each EV. For example, when there are multiple roads between RSU A and B, FADEC can use the road that takes the longest time to travel to estimate $t_{A\to B}$. Second, FADEC can tolerate the overestimation of $t_{A\to B}$ and t_B at the price of increased storage requirement. Using the mobility-prediction approach [20], if the prediction is not accurate and the EV does not move towards the predicted next RSU, the EV has to run the key exchange protocol again to establish a new session key with the RSU, which could consume several seconds of valuable contact time with the RSU.

4.3.2 Establishing Session Key K^u_e with the Utility

An EV establishes K^u_e using JFK, but only after it has established K^r_e with the RSU. Since the EV cannot directly communicate with the utility, it has

to send the JFK messages to an RSU, and the RSU will relay the messages to the utility. Since the EV has already established K_e^r with the RSUs, it will sign the JFK messages using K_e^r before sending them to the RSU, and the RSU will verify the signature before relaying the messages. When the utility replies, the RSU will also sign the reply using K_e^r, and then send it to the EV.

4.3.3 Prioritizing Key Establishment Messages

When an EV is sending or receiving JFK messages to establish keys, other EVs that have completed their key establishment might be sending application messages (e.g., content delivery) at the same time. The application message traffic can have a non-negligible impact on the key establishment duration, as the RSU queue is likely to have many more application messages than JFK messages. Without careful design, the processing of JFK messages could be delayed indefinitely in the RSU.

We solve this problem by having each RSU maintain two queues: a JFK queue that stores only messages related to the JFK protocol, and a normal data queue. An RSU prioritizes the processing of JFK messages, and will start processing messages from the data queue only when the JFK queue is empty. In this way, key establishment messages will not be delayed because of application messages that have arrived earlier. In our implementation, the JFK queue employs the First-In First-Out (FIFO) scheduling policy while the data queue employs the Earliest Deadline First (EDF) policy.

4.4 Security Analysis

4.4.1 Replay Attack

The attacker could replay an EV's message to an RSU to confuse the billing system, or could replay an RSU's message containing pricing information to mislead nearby EVs. Replay attacks can be prevented by either including a timestamp or a nounce in every message exchanged to ensure freshness.

4.4.2 DoS Attack

The attacker could flood an RSU with fake key establishment messages (DoS against authentication) or with fake reports (DoS against reporting). In the

first case, the DoS attack is mitigated by the use of DoS-resistant JFK as the key exchange protocol. In the second case, since FADEC uses HMAC authentication to ensure fast signature verification, the effectiveness of a DoS attack is greatly reduced.

4.4.3 Bruteforce Attack

The attacker could launch a bruteforce attack by collecting messages and corresponding signatures, and using bruteforce algorithm to recover the session key. However, it is computationally infeasible to recover the session key from a HMAC signature, no matter how many signatures signed by the same key are exposed to the attacker. With more message-signature pairs, the attacker has better chance to guess the correct session key. To limit session key exposure, conventional approaches allow the EV to establish a new session key with every RSU. However, if the EV sends data frequently, the attacker might still be able to collect enough message-signature pairs of the same session key. On the other hand, FADEC allows EV to decide when to expire the current session key according to the amount of data signed using the key: if there are small number of messages signed using the current session key, our key-dissemination approach allows the EV to continue using the key with the next RSU; if the EV has signed a large number of messages using the current session key, it can re-establish a new session key with the current RSU.

4.4.4 Wireless Jamming

The attacker could also attempt to jam the wireless channel between EVs and RSUs. If the attacker succeeds, not only would the FADEC authentication messages be blocked, but all wireless communication between EVs and RSUs would be impossible. Wireless jamming is a general threat to wireless communication and is out of our scope.

4.4.5 Compromising RSUs

The attacker could attempt to compromise one or multiple RSUs and obtain the session key K_e^r shared between EV e and RSU. Since FADEC allows EV e to use the same session key K_e^r with all RSUs, once the attacker obtains the key K_e^r, he could pretend to be EV e and convince other RSUs to relay its message to the utility. However, since the utility and the EV authenticate

each other using another key K_e^u, the utility can easily recognize if the message comes from the attacker, and can further inform the subsequent RSUs that the session key K_e^r has been compromised. The next RSU then expires the compromised key K_e^r, and re-negotiates a new session key with the EV if necessary.

4.4.6 Man-in-the-Middle (MITM) Attack

During the key establishment phase, MITM attack is impossible since JFK messages are digitally signed, and the attacker cannot impersonate any party establishing K_e^u or K_e^r. In particular, the attacker cannot tamper the key establishment messages between EV e and the utility, even if the messages are relayed by a compromised RSU controlled by the attacker. After K_e^u and K_e^r are established, a compromised RSU cannot impersonate an EV since K_e^u is only shared between the EV and the utility, and is not known by any RSU.

4.4.7 Impersonation Attack

Since EV e and the utility authenticate each other using session key K_e^u known only by the utility and EV e, the only way for the attacker to convince EV e to accept a forged message from the utility is by compromising the utility itself and obtaining K_e^u, which is impossible according to our attack model. Similarly, the attacker can only impersonate EV e by actually compromising the EV. Since K_e^u is not stored at any RSU, although the attacker may be able to obtain session key K_e^r shared between EV e and the RSUs by compromising RSUs, the attacker cannot forge any message between the EV and the utility.

4.4.8 EV Misreporting

FADEC does not provide any semantic guarantee on the correctness of the reports sent by EVs. Although an EV cannot pretend to be another EV, it can still report less energy received than actual in order to reduce payment. The detection of misreporting is out of our scope.

4.5 Performance Evaluation

We simulate road traffic on a 4-lane single-direction straight road segment of 3km, with a total of 5 RSUs deployed evenly along the road segment, at distances 0.3, 0.9, 1.5, 2.1, and 2.7 km from the start of the road segment. We use SUMO [21] to generate mobility traces from a congested traffic flow with 7284 EV/hour where the vehicles travel at a maximum speed of 75 km/h (46.9 mph), which has been observed on I-10 westbound [22]. We use the mobility trace of 300 EVs as they traverse the 3 km long road segment; every EV starts from a randomly chosen lane, and the simulation stops when all EVs have left the road segment. In order to evaluate the system in steady state, we show results for EVs 100 to 199, i.e., we discard the results of the first and the last 100 EVs.

We simulate a backbone connection between the utility and each RSU, and between each pair of neighbor RSUs. The propagation delay between the utility and each RSU is set to 100 ms, and the delay between neighbor RSUs is set to 1 ms. We use the Veins [23] simulator to simulate IEEE 802.11p MAC layer behavior. We use the default 802.11p settings from the Veins simulator for both the RSU and the vehicles; the RSU can communicate with vehicles within approximately 500 meters. For each pair of neighbor RSUs A and B we set $\hat{t}_{A\to B} = 120$ sec, and for each RSU A we set $\hat{t}_A = 120$ sec.

We evaluate FADEC in two scenarios with different assumptions on the computational resource available to the EV and the RSU. In the *resource rich* scenario, we assume the EV and the RSU have a strong CPU to sign messages and to verify signatures; in this scenario the signing and verification using digital signature both take 20 ms. In the *resource constrained* scenario, the EV and the RSU hardware have less computational power; in this scenario digitally signing a message and verifying a digital signature both take 200 ms.

IEEE 1609.2 [24] requires ECDSA to use either NIST P-224 or P-256 elliptic curve. The resulting signature lengths are 448 bits and 512 bits respectively. In our simulation we choose ECDSA with P-224 curve, which generates shorter signatures. We use JFK with 2048-bit RSA field and 2048-bit DH field to generate 224-bit session key, and HMAC-SHA-1 as the MAC implementation to compare with ECDSA. Note that the message overhead of JFK applies only once per EV, since an EV runs JFK only when it first enters the charging section. Both HMAC-SHA-1 with 224-bit session key and ECDSA with P-224 curve provide 112-bit security strength, which is acceptable today [15].

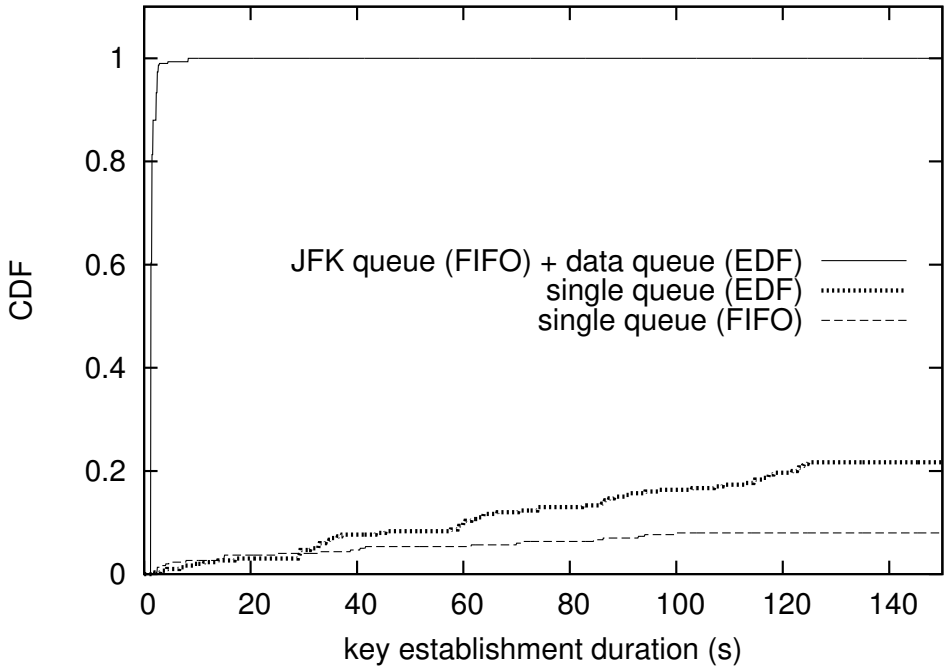

Figure 4.3: Key establishment duration of the first 100 EVs in the resource constrained scenario with different RSU queue management strategies.

In all our simulations the EVs generate 1024 bits of information per second. Unless otherwise noted, each EV sends a report to the utility every 5 seconds containing all information since the generation of the last report. The deadline for each report is set to be 5 seconds after its creation time, since after 5 seconds the EV will generate a new report.

4.5.1 Key Establishment

We first consider the time it takes for an EV to establish its keys. Recall that an EV e first establishes K_e^r with the RSU, and then establishes K_e^u with the utility. The successful establishment of K_e^u thus implies the establishment of K_e^r.

A natural question is whether it is necessary to prioritize key establishment message processing. As alternatives, we consider two solutions: (i) the RSU maintains a single data queue for both EV reports and key establishment messages and employs FIFO scheduling policy; (ii) the RSU maintains a single data queue but applies the EDF scheduling policy. The deadline for a key establishment message is set to 1 second.

In Figure 4.3 we show the distribution of the time it takes for an EV to establish keys with both the RSU and the utility in the resource constrained

scenario. We use results from the first 100 EVs to illustrate how the system reaches its stable state. The results show that maintaining only one queue for both key establishment messages and data messages does not guarantee the success of key establishment for all EVs. Using a single FIFO queue, only 8% EVs finish their key establishment, and although using EDF scheduling helps, still less than 30% of the EVs can complete their key establishments.

Prioritizing key establishment messages by maintaining a separate queue for JFK greatly reduces the key establishment duration. Over 80% EVs establish K_e^u within 1.7 seconds even in the resource constrained scenario. In the worst case the key establishment takes 8.3 seconds. Note that an EV performs key establishment only once, and uses the same K_e^r (K_e^u) with every RSU (the utility). The one-time cost of 8.3 second is small compared to the time scale in a dynamic EV charging scenario (about 144 seconds in our case). These results show that prioritization is essential for successful key establishment in FADEC when computational resources are scarce.

4.5.2 Reporting Period

One point of uncertainty in terms of the communication needs for dynamic charging is the reporting period. At one extreme, the EV could accumulate information and could send one large report containing all information when leaving the charging pad; at the other extreme, the EV could send reports very frequently, with each report containing only a small amount of information. We therefore start with investigating how often an EV could send reports to the utility with and without FADEC. We consider that the EVs send periodic reports every t seconds, where t ranges from 5 to 9, and a report is delivered successfully if it arrives at the utility within t seconds. Each report contains all information generated by the EV since the last report sent. With a large value of t the EVs send reports less often, but each report is larger as it contains more information.

In the resource rich scenario, both FADEC and ECDSA achieve delivery ratio close to 1. In Figure 4.4 we show the delivery ratio as a function of the reporting period in the resource constrained scenario. The curves show the delivery ratio of reports averaged across all EVs, and the error bars indicate the 5th and the 95th percentiles. We can observe that FADEC is almost insensitive to the reporting period and achieves a delivery ratio close to 1. ECDSA, on the other hand, achieves a very low delivery ratio when reports are sent frequently, even though EDF scheduling is used in the RSU. The reason is that the RSU cannot perform the verification needed by

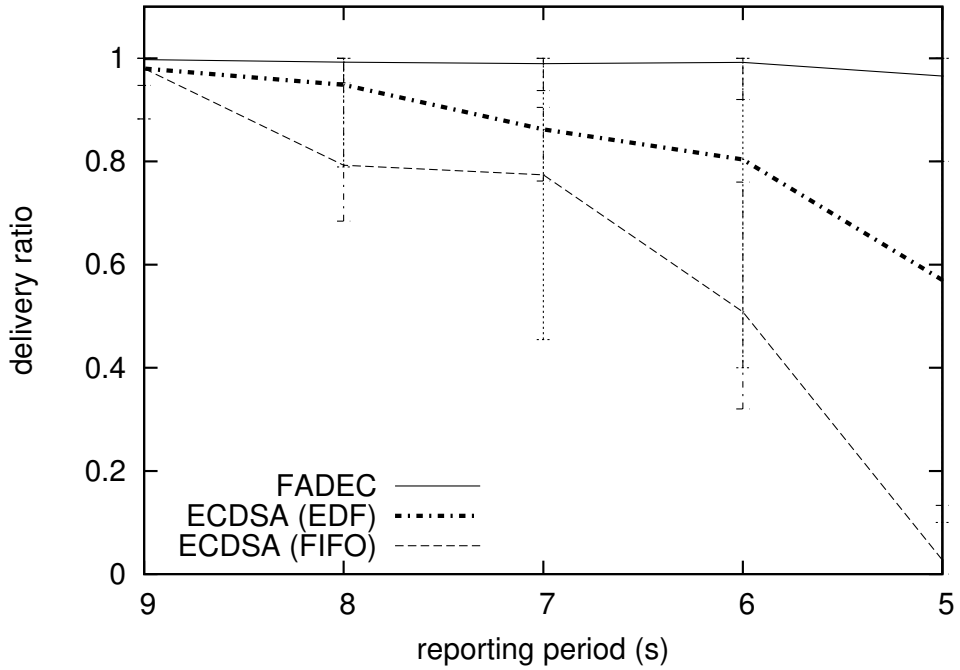

Figure 4.4: Report delivery ratio under different reporting period in the resource constrained scenario.

ECDSA at the rate at which reports arrive. As a result, the RSU data queue keeps increasing, and earlier reports miss the deadline. The delivery ratio of FADEC is not only higher, but it is also more stable across all EVs; the 5th and the 95th percentiles are close to the average, whereas the percentile intervals for ECDSA are rather wide. In the following we use ECDSA with EDF for comparison.

4.5.3 Reliability and Throughput

Achieving consistently high data throughput is important for dynamic EV charging, since it allows the utility to obtain up-to-date information about the EV status. In our scenario where all EVs send reports at the same frequency, throughput is proportional to the delivery ratio.

In Figure 4.5 we show the distribution of the delivery ratio of reports from each EV for the two scenarios. Using FADEC, most EVs are able to achieve a delivery ratio close to 1 in both scenarios. Using ECDSA results in lower delivery ratios, especially in the resource constrained scenario, where only 57% of the reports are delivered successfully on average. The reason is that ECDSA's large signing and verification overhead makes the RSU data queue grow quickly, and most reports miss their deadlines even using EDF

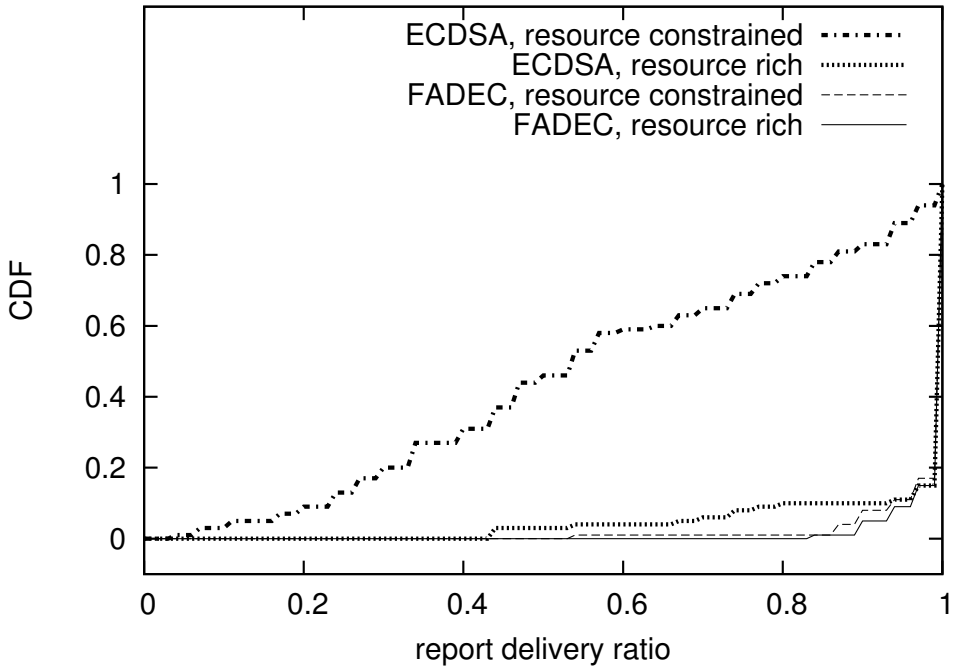

Figure 4.5: Distribution of report delivery ratio.

scheduling.

4.5.4 Delay

In Figure 4.6 we plot the distribution of the delay of all reports that arrived at the utility within their deadlines. This is an important metric for our evaluation, since a shorter delay means the utility could receive reports from the EV sooner and would thus have better knowledge of the current charging profile of the EVs, and the instantaneous demand.

The delay includes the time taken by the EV to sign the report, the delay due to 802.11p channel access and data transmission, the time taken by the RSU to verify the signature, backbone network delay, and the time taken by the utility to verify the signature. FADEC achieves almost the same delay with an average of 0.117 second in both scenarios. By design, FADEC is insensitive to the increased cost of digital signature operations in the resource constrained scenario, since once the session keys are established, signing a message or verifying a signature takes only one or two hash operations according to HMAC. On the other hand, the average delay of ECDSA in the resource rich scenario is 0.180 second, and increases to 4.805 seconds in the resource constrained scenario. In the resource constrained scenario, the time to sign a message and to verify a signature using ECDSA significantly

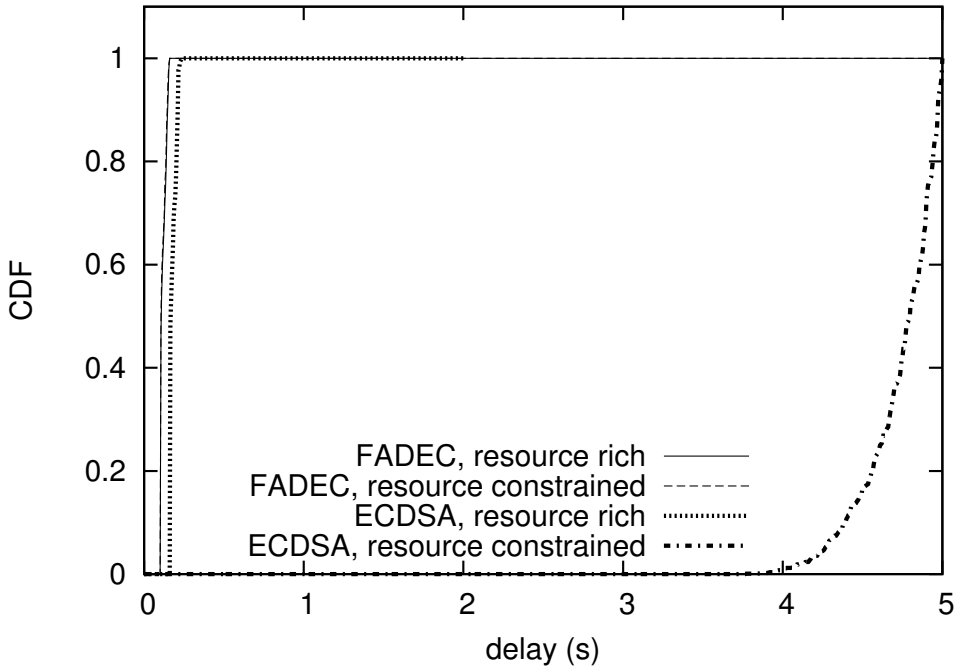

Figure 4.6: Distribution of delay of successfully delivered reports.

increases. This greatly affects the delay of ECDSA.

4.6 Extending FADEC for Anonymous Reporting

Although FADEC enables efficient real-time reporting between EVs and the utility, the design of FADEC as we have described so far does not fully protect the EV's location privacy. In particular, since the EV uses the same session to authenticate with a series of RSUs, an observer can eavesdrop on the wireless communication channel and track the EV's positions by following the use of the same session key. This problem can be mitigated by having the EV periodically negotiate a new session key with the RSU. One example design follow this approach is TACK [25], which divides the road into segments. Within the same road segment the EV uses the same session key, which provides short-term linkability; and the EV uses different session keys at different road segments to achieve long-term unlinkability. Another part of the FADEC design that compromises the EV's location privacy is the use of JFK as the key establishment protocol. JFK assumes the deployment of PKI and uses the EV's long-term public key for authentication. Thus, although the EV could periodically re-negotiate a new session key, if JFK is used for each key negotiation, an outside observer can still link the old

session key and the new session key to the same EV by observing that the same long-term public key is used in the JFK negotiations.

We observe that in many real-time reporting scenarios, the utility is only interested in the aggregate result from many EVs' reports rather than the report concerning an individual EV's identity. For example, the utility may request EVs within a certain area to report their current battery State-of-Charge (SoC) in order to predict the charging demand in the near future. In this case, the utility is interested in questions such as how many EVs within this area have battery SoC below 30%, instead of the current battery SoC of a particular individual EV. From the perspective of authentication and reporting, the utility only needs to verify that the reports are indeed submitted by legitimate EVs (as opposed to forged by an attacker using a laptop) but does not need to know the true identity of each particular EV. In particular, the EV should main anonymous during the reporting process. FADEC can be easily extended to enable anonymous reporting. In its current design, the session key can be linked to the EV's true identity because FADEC uses JFK for key establishment, and JFK uses PKI-based authentication which reveals the EV's true identity. To enable anonymous report, anonymous credentials [26, 27, 28] can be used to authenticate the EV during the key establishment process: the EV spends an anonymous credential and binds the anonymous credential to the key establishment session (e.g., [29]) instead of using PKI-based authentication.

4.7 Related Work

Host Identity Protocol (HIP) [30] is a popular solution for micro-mobility. Whenever the EV changes its network location (e.g., moves into the range of a new RSU), it sends an UPDATE message to notify the rendezvous server about its new network location. Despite efforts [31] to reduce control signaling and to simplify the update procedure, HIP-based approaches still incur non-trivial handover latency. The proposed FADEC mechanism differs from HIP-based approaches in that it incurs no handover latency: the next associated RSU always obtains the session key before the EV enters its range, and the EV continues to use the current session key with the next associated RSU. Zhu et al. [20] suggest a prediction-based approach, where the current RSU predicts the next RSU that the EV will encounter, and pre-establish a session key between the EV and its next associated RSU. The drawback of this approach is that the performance highly depends on the accuracy of EV

mobility prediction. If the current RSU does not correctly predict the next associated RSU, the EV itself would have to re-establish a new session key with the next RSU. FADEC, on the other hand, does not predict individual vehicle mobility, but only uses aggregate traffic statistics such as the average speed of vehicles along a road segment, which can be easily obtained from historical data.

In dense traffic area, an RSU would need to simultaneously verify packets from multiple vehicles. This motivates several batch authentication designs [32, 33, 34, 35, 36, 37, 38, 39, 40], where the packet is first batched without verification, and the batch of packets is verified either periodically or when the batch accumulates to a certain size. In this way, batch authentication reduces the total computation overhead. However, since the packets are not verified immediately, batch authentication may not be suitable for real-time applications.

Authentication based on one-time signatures [14, 13, 41] has also been considered in vehicular networks. Due to its fast and lightweight verification, one-time signature is particular attractive in broadcast scenarios, e.g., an RSU broadcasting electricity price information to nearby EVs. VAST [42] combined an improved version of TESLA one-time signature [14] with ECDSA [43] to provide flexible and efficient authentication for vehicular network. Hsiao et al. [12] showed that one-time signatures can be further optimized if the future content to be signed can be predicted.

4.8 Conclusion

In this chapter, we have presented FADEC, authentication for dynamic electric vehicle charging. FADEC lets EVs establish symmetric keys with the RSUs and the utility, and achieves fast signing, fast verification, fast handoff authentication, and low communication overhead. Our simulations have shown that FADEC with EDF scheduling obtains very close to 1 report delivery ratio and small delay in both resource rich and constrained scenarios, and is more suitable for dynamic electric vehicle charging than ECDSA. We have also described how to extend FADEC to provide anonymous reporting by using anonymous credentials to establish session keys.

CHAPTER 5

PORTUNES

Dynamic charging requires communication between the EVs and the pads. The EV needs to inform each wireless 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. In addition, the EV and the charging pads must be able to verify the identity of each other upon exchanging information in order to defeat any malicious attempt to impersonate the EV or the charging pads.

Designing an authentication scheme for EVs to authenticate with charging pads is challenging. 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, and the authentication must complete within no more than several milliseconds so that the rest of the contact window can be used to actually charge the EV. Since there are many short charging pads in a dynamic charging 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 [12] and is infeasible in this scenario. One-time signature schemes [13, 14] 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 [44] that requires multiple message exchanges is less likely to succeed due to packet losses in vehicular networks [45].

In this chapter we describe Portunes, a privacy-preserving authentication protocol that allows fast authentication between EVs and charging pads, and provides location privacy through using pseudonyms. 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 this chapter is organized as follows: in Section 5.1, we describe the system model; in Section 5.2, we present the Portunes protocol; in Section 5.3, we analyze various security and privacy aspects of Portunes; in Section 5.4, we present evaluation results; in Section 5.5, we review related work; and we conclude this chapter in Section 5.6.

5.1 Model and Assumptions

We consider a system that consists of utility, *pad owners* (PO) and electric vehicles (EVs).

5.1.1 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. For clarity we assume the charging pads are numbered $1, 2, 3, \ldots$, and the EV always encounters the charging pads in ascending order.

5.1.2 Communication Model

We assume that the utility and the PO are connected through a high speed 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. We also assume that each charging pad is able to communicate with its predecessor and successor charging pads through PLC. Finally, each EV can communicate with the utility 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.

5.1.3 Time and Location Information

We assume that the utility, 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., [46]), and learns its GPS coordinates l_p from the PO.

5.1.4 Billing Model

Portunes is designed for the subscription-based billing model introduced in Section 2.4. We refer the reader to Section 2.4 for a detailed description of the billing model.

5.1.5 Security and Attack Model

We assume deployment of a PKI. The utilities, the POs, and each EV have a pair of public and private keys. Each utility knows the subscribing EVs' public keys, and each EV also knows its utility's public key. The utilities know the public keys of the POs and vice-versa. In addition, a PO P shares two symmetric keys $K^E_{P,p}, K^A_{P,p}$ with each of its charging pads p. Each utility

C also shares a one-way function $f_{C,P}$ with each PO P (and its charging pads).

We assume the attacker can eavesdrop on the wireless communication between the EV and the infrastructure such as the utility, the PO, and the charging pads. In particular, the attacker could capture the message sent by an EV and replay the message somewhere else. 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, and that the attacker cannot compromise the utility, the pad owner, or any charging pad and obtain their secret keys. Portunes focuses on defending against impersonation attacks, where the attacker pretends to be the EV, the charging pads, the PO, or even the utility in order to benefit. One example is that the attacker, who charges his own EV using dynamic charging, pretends to be another EV to evade payment.

In Table 5.1 we summarize the notations used in the chapter. 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.

5.2 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 two phases: key pre-distribution and authentication. In the key pre-distribution phase, the utilities 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 utilities 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 Figure 5.1 we show the message exchange. Note that msg 2 is between the PO and each charging pad, and msg 5 and msg 6 are between the EV and each charging pad.

I_e	the permanent identity of EV e.
π	pseudonym assigned by the utility to an EV.
Π	the set of all pseudonyms.
$f_{C,P}$ (or f)	collision-free one-way function shared between utility C and PO P.
$K^E_{f(\pi)}, K^A_{f(\pi)}$	session keys assigned by the utility to EV with pseudonym π.
$\text{EtM}^{K^A}_{K^E}$	Encrypt-then-MAC with encryption key K^E and MAC key K^A.
$\mathcal{K}_{C,P}$ (or $\mathcal{K}$)	the key set $\{(\pi, K_{f(\pi)}) : \pi \in \Pi\}$ of all index-key pairs sent by utility C to PO P.
$K^E_{P,p}, K^A_{P,p}$	symmetric keys shared between pad p and PO P.
$K(m)$	AES encryption of message m using symmetric key K.
$\{m\}_{A\to B}$	sign the message m using A's privacy key, then encrypt m and the signature using B's public 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.

Table 5.1: Notations

5.2.1 Key Pre-distribution Phase

The key pre-distribution phase occurs every night, when there is little road traffic. Utility C generates the pseudonym set Π and the corresponding indexed key set

$$\mathcal{K} = \{(f(\pi), K^E_{f(\pi)}, K^A_{f(\pi)}) : \pi \in \Pi\} \tag{5.1}$$

using a collision-free one-way function f, where $K^E_{f(\pi)}$ is for message encryption/decryption and $K^A_{f(\pi)}$ is for MAC computation. f is one-way in that it is infeasible to compute π given $f(\pi)$. Since we assume the pseudonym 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 $\pi \in \Pi$, utility C

[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.

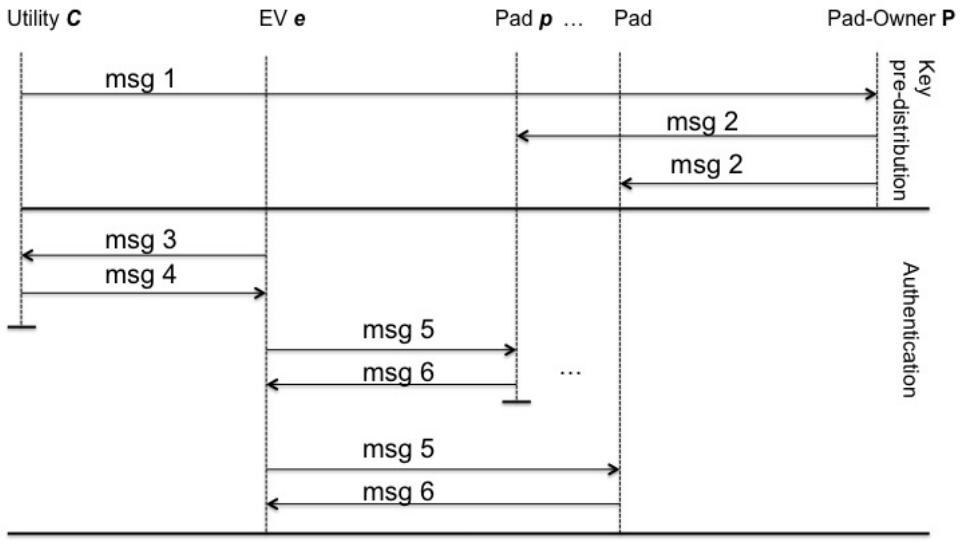

Figure 5.1: Portunes protocol overview with utility C, EV e, pad-owner P and charging pad p. Messages 1 to 6 are specified in equations (1)-(6), respectively.

sends

$$\text{msg } 1 : \{f(\pi), K^E_{f(\pi)}, K^A_{f(\pi)}, t_C\}_{C \to P} \tag{5.2}$$

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

When receiving msg 1, the PO disseminates the learned index-key tuples $(f(\pi), K^E_{f(\pi)}, K^A_{f(\pi)})$ to the charging pads by sending the message

$$\text{msg } 2 : \text{EtM}^{K^A_{P,p}}_{K^E_{P,p}}(f(\pi), K_{f(\pi)}, t_P) \tag{5.3}$$

to each pad p, where t_P is the current timestamp generated by the PO. This message is first encrypted with key $K^E_{P,p}$, and the MAC on the ciphertext is computed with key $K^A_{P,p}$. In the end, each charging pad learns the entire key set $\mathcal{K}$.

5.2.2 Authentication with Utility and Charging Pads

Upon entering a charging section, EV e authenticates with utility C to obtain a pseudonym π and the session keys $K^E_{f(\pi)}, K^A_{f(\pi)}$. As the EV moves within the charging section, it uses π and the session keys to encrypt the message and authenticate with each charging pad it encounters.

EV-Utility Authentication

In order to authenticate with the utility, EV e sends

$$\text{msg } 3 : \{I_e, t_e\}_{e \to C} \tag{5.4}$$

to utility 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.

When receiving msg 3 from EV e, utility C decrypts the message and verifies the EV's digital signature. 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

$$\text{msg } 4 : \{I_e, t_e, t_C, \pi, K^E_{f(\pi)}, K^A_{f(\pi)}\}_{C \to e} \tag{5.5}$$

back to EV e, where $K^E_{f(\pi)}$ and $K^A_{f(\pi)}$ are the encryption and authentication keys with index $f(\pi)$, t_e is the timestamp received in msg 3, and t_C is the utility'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 utility C to ensure its authenticity.

EV-Pad Authentication

Once on the charging section, in order to authenticate with a charging pad within range, EV e periodically broadcasts the beacon

$$\text{msg } 5 : \mathit{beacon} = (\pi, \mathrm{EtM}^{K^A_{f(\pi)}}_{K^E_{f(\pi)}}(C, \pi, t_e, \hat{l}_e(t_e), \mathit{req})), \tag{5.6}$$

where C is the utility that assigned the pseudonym π and the session keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ 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 15 ms.

The pseudonym π in plaintext is used by the pad to locate the corresponding session keys. When pad p receives the beacon, it uses the mapping f shared with utility C to compute $f(\pi)$. It then verifies the MAC on the ciphertext using key $K^A_{f(\pi)}$. If the MAC verification succeeds, the ciphertext

is not tampered. Pad p then decrypts the ciphertext using key $K^E_{f(\pi)}$, and 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 5.2.3 and 5.3, respectively.

If all verifications succeed then pad p will switch on and charge the EV. At the same time it removes the corresponding keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ 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

$$\text{msg } 6 : \text{EtM}^{K^A_{f(\pi)}}_{K^E_{f(\pi)}}(\pi, t_e, t_p, l_p, ack) \tag{5.7}$$

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.

5.2.3 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 [47], 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

$$\|l_p - l_e(t)\| < \sqrt{r^2 - h^2} + \bar{v} \cdot \tau, \tag{5.8}$$

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

$$\hat{l}_e(t') = \hat{l}_e(t) + \vec{v}_e(t) \cdot (t' - t) \tag{5.9}$$

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.

5.2.4 Implicit Authentication

We say an EV is explicitly authenticated by a charging pad if the charging pad successfully receives and verifies the EV's beacon as described in section 5.2. Due to the unreliable nature of the wireless channel, the EV may still fail to explicitly authenticate with the charging pad , as we will see in Section 5.4. In order to increase the probability of overall successful authentication, in this section we propose an implicit authentication protocol.

[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 15 ms, i.e., $t' - t = 15$ ms, the location error introduced by speedometer inaccuracy is at most (8 km/h·15 ms =) 0.03 meter.

One approach to increase the probability of overall successful authentication would be to let charging pad p that explicitly authenticates EV π at time t to send a *witness message* to the next m charging pads $p+1, p+2, \ldots, p+m$. With a properly chosen m, it is safe to assume that between the moment t when EV π authenticates with pad p and the future moment t' when EV π reaches pad $p+m$, only EV π is moving from pad p to pad $p+m$. Then, for each $p < q \leq p+m$, charging pad q can implicitly authenticate the first EV seen after time t as EV π if it receives the witness message from p, without directly verifying the EV's beacon. Note that only the charging pads that explicitly authenticate the EV can generate witness messages, whereas a charging pad that implicitly authenticates the EV can only forward a previous witness message. This guarantees that the EV must explicitly authenticate with at least one of every $m+1$ charging pads.

Whenever a charging pad p explicitly authenticates an EV as described in Section 5.2, it forwards a witness message (p, π, v, t) to the next m charging pads $p+1, \ldots, p+m$, where π is the authenticated EV's pseudonym, v is the EV's speed, and t is the witness time. The witness pad can either learn the EV's speed if the EV includes its speed in the beacon, or can measure the EV's speed with sensors. When pad q receives the witness message (p, π, v, t) from pad p, it computes the estimated time-of-arrival $eta = t + \frac{d(q-p)}{v}$, where $d = \lambda + \delta$ is the distance between the wireless devices of two neighbor charging pads. If pad q receives two messages (p, π, v_1, t_1) and (p', π, v_2, t_2) about the same EV π where $p' < p < q$, it uses the message (p, π, v_1, t_1) from pad p to compute eta, and discards the message from pad p'. If at time t' where $t' < eta$, pad q learns that some EV is above it (even if pad q fails to receive the EV's beacon, it can still learn its presence through pressure sensors), pad q implicitly authenticates the EV as π and starts charging its battery.

5.3 Security and Privacy Analysis

If the attacker compromises the utility 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. Portunes assumes that the infrastructure, such as the utility, the PO, and the charging pads, are trusted, and the above attacks are out of the scope of this chapter.

An attacker driving an EV may capture the beacon (msg 5) sent by EV e to pad p by either following the victim EV e or with the help of receivers previously deployed by the attacker along the charging section. Once the attacker captures the beacon, he could replay the beacon to a pad p' and impersonate EV e. 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 decrypt without the key $K_{f(\pi)}$.

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.

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.

In Portunes, the utility is able to learn the mapping between the EV's true identity and its pseudonym because in the EV-utility authentication step, the EV authenticates itself using PKI with long-term public key in order to obtain the pseudonym from the utility. To enhance the location privacy of EVs, anonymous credentials can be used to replace PKI authentication so

that the utility does not learn the mapping between the EV's pseudonym and its true identity. In particular, the utility would issue a number of single-use anonymous credentials to the EV at the beginning of the billing cycle. During the EV-Utility authentication step in Portunes (i.e., msg 3 in Section 5.2.2), instead of using PKI-based authentication, the EV would simply spend an unused anonymous credential to prove to the utility that it is a legitimate EV, e.g., the utility could send a random challenge to the EV and the EV would reveal an unused anonymous credential and bind the double-spending equation to the random challenge similar to the price validation part of Janus described in Section 6.4.2.

5.4 Evaluation

In this section we present evaluation result on the performance, overhead, and reliability of Portunes.

5.4.1 Authentication Speed

We implemented Portunes on Raspberry Pi 2 Model B [48] using Crypto++ 5.6.2. The RaspberryPi features a 900 MHz Quad-core CPU and 1 GB RAM, and costs $35 (USD) at the time of writing. For comparison we also implemented Elliptic Curve Digital Signature Algorithm (ECDSA) [43], which is recommended by the current IEEE 802.11p standard for authenticating vehicular communication, on the same platform. In Figure 5.2 we compare the generation and verification time of the beacon (msg 5) using Portunes and using ECDSA. We use AES with CFB mode and a 128-bit key for symmetric encryption in Portunes, and use ECDSA on P-224 curve, which results in a 448-bit signature. Both Portunes and ECDSA provide 112-bit security strength in this setup. We assume the EV's true identity and pseudonym are both 64-bit.

Portunes takes 0.07 ms to generate a 100 byte beacon, and 0.02 ms to verify the beacon. The generation and verification times increase to 0.15 ms and 0.11 ms, respectively, as the beacon size increases 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 9 ms to generate a signature, and over 14 ms to verify a signature, i.e., almost two orders of magnitude more.

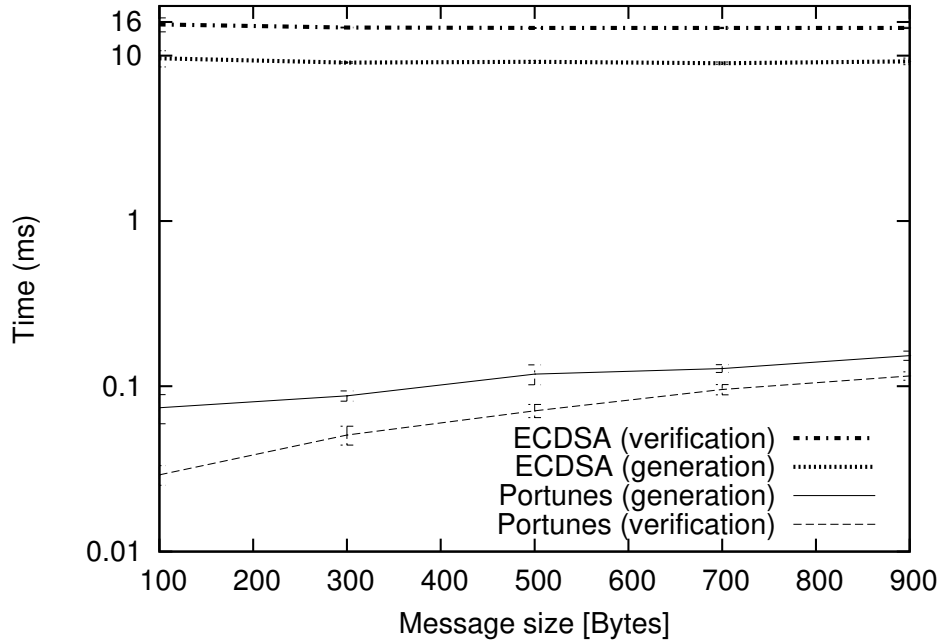

Figure 5.2: Generation and verification time of beacon (msg 5) using Portunes and ECDSA vs. message size. Error bars indicate 95% confidence intervals.

5.4.2 Reliability

Since an EV must authenticate with the charging pad before it can be charged, it is important to ensure a high authentication success probability for the range of EV speeds of interest through choosing an appropriate beacon broadcast frequency. In the following we derive a model to support choosing a good beacon broadcast frequency.

For simplicity we consider an EV e that moves at constant speed v over the charging section. Varying speed can be included in the model at the expense of increased complexity. We denote by f the frequency at which the EV broadcasts its beacon, and by x^i the location of the EV upon the i^{th} broadcast, thus, $x^i = x^1 + (i-1)\frac{v}{f}$. We denote by x_p the location of charging pad p, and without loss of generality we let $x_1 = 0$ (i.e., all locations are relative to that of the first charging pad). We define the first broadcast ($i = 1$) to be the first broadcast within range of charging pad $p = 1$. Note that if the EV starts to send beacons without knowledge of the pads' locations then the location x^1 of the first broadcast is uniformly distributed on $(-\sqrt{r^2 - h^2}, \sqrt{r^2 - h^2})$. Finally, we denote by d_p^i the distance between the communication device of the EV and that of charging pad p

upon the i^{th} broadcast,

$$d_p^i(x^1) = \sqrt{h^2 + (x_p - x^1 + (i-1)\frac{v}{f})^2}. \tag{5.10}$$

Let us denote by P_t the transmit power used by the EV to transmit the beacons, and by $PL(d)$ the air path loss as a function of the distance d, which is assumed to be quadratic. For given receiver bandwidth B, noise power N and target bit rate f_b we can compute the per bit energy to noise ratio as

$$\frac{E_b}{N_0} = P_t - PL(d) - N - \frac{f_b}{B} \text{ in dB}, \tag{5.11}$$

and assuming that the radio channel is subject to additive white Gaussian noise (AWGN) we can compute the bit error rate $\beta(d)$ as a function of the distance d between the EV and the charging pad as

$$\beta(d) = \frac{1}{2}\text{erfc}(\sqrt{\frac{E_b}{N_0}}) \tag{5.12}$$

.

In order to combat bit errors we assume that broadcasts are protected by a channel code with rate $\frac{n-c}{n}$, where n is the total number of bits in a beacon, and c is the number of redundant bits. Assuming a binary erasure channel, such a code is sufficient for correcting up to c bit errors among n bits. Since the horizontal displacement of the EV between transmitting two consecutive bits is very small, we can consider that all bits of a broadcast are transmitted at the same distance from the receiving pad. Assuming an i.i.d. loss process, we can thus compute the probability $\gamma(d)$ that a broadcast from the EV is received successfully over a distance d as

$$\gamma(d) = \sum_{j=0}^{c} \binom{n}{j} \beta(d)^j (1-\beta(d))^{n-j} \tag{5.13}$$

In order to obtain a lower bound on the authentication probability, we make the assumption that the success probability $\gamma(d)$ is negligible when $d > \sqrt{r^2 - h^2}$, and we denote by p_i the charging pad closest to the location of the EV upon broadcasting beacon i. Let A denote the event that the EV successfully authenticates with the charging pad. Using the above we can now express the cumulative probability of successful authentication conditional on the relative location x of the EV and on the location of the first

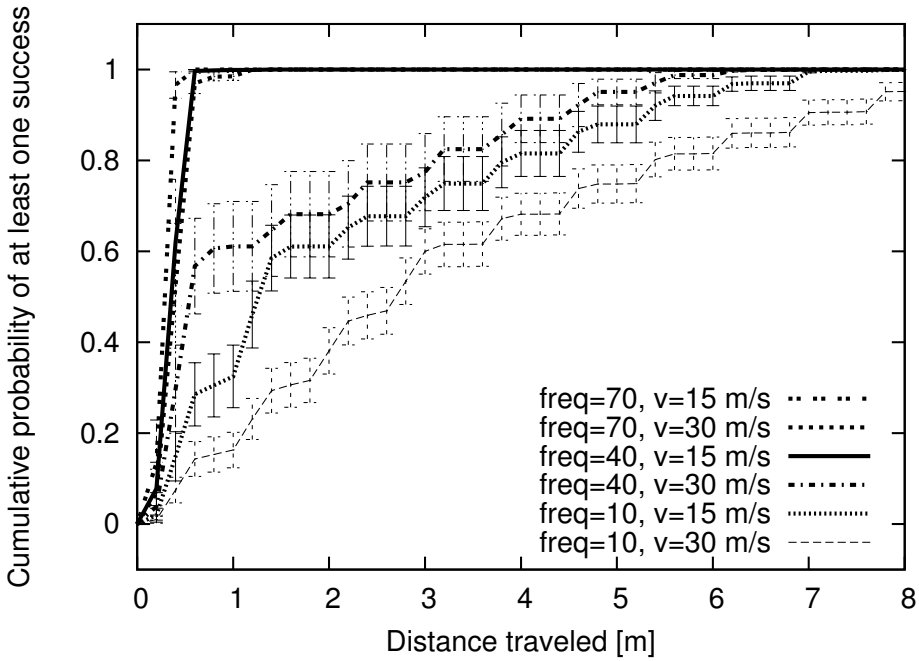

Figure 5.3: Cumulative probability that the EV has made at least one successful authentication as a function of the distance traveled.

beacon broadcast as

$$P(A|x, x^1) = \sum_{i=1}^{\lfloor (x-x^1)/(v/f) \rfloor} \gamma(d^i_{p_i}(x^1)) \prod_{j=1}^{i-1} (1 - \gamma(d^j_{p_j}(x^1))), \tag{5.14}$$

and can use the law of total probability to compute $P(A|x)$ using the distribution of x^1.

In the following we show results for $r = 0.5$ m, $h = 0.3$ m, $\frac{c}{n} = 1/8$, and $n-c = 512$ bits. In Figure 5.3 we plot the cumulative probability that the EV has made at least one successful authentication as a function of the distance traveled. The figure shows that a broadcast frequency of 10 is insufficient to achieve a good authentication probability; a medium broadcast frequency of 40 is sufficient when the EV is moving at a low speed of 15 m/s; but a broadcast frequency of 70 guarantees high authentication success probability even at high speeds. The almost step-wise increase of the curves is due to the regular placement of the charging pads.

In Figure 5.4 we plot the probability of at least one successful authentication as a function of the inter-beacon distance, i.e, $\frac{v}{f}$, for various values of the distance traveled. We observe that the curves corresponding to a larger total travel distance are above the lines that correspond to a smaller total

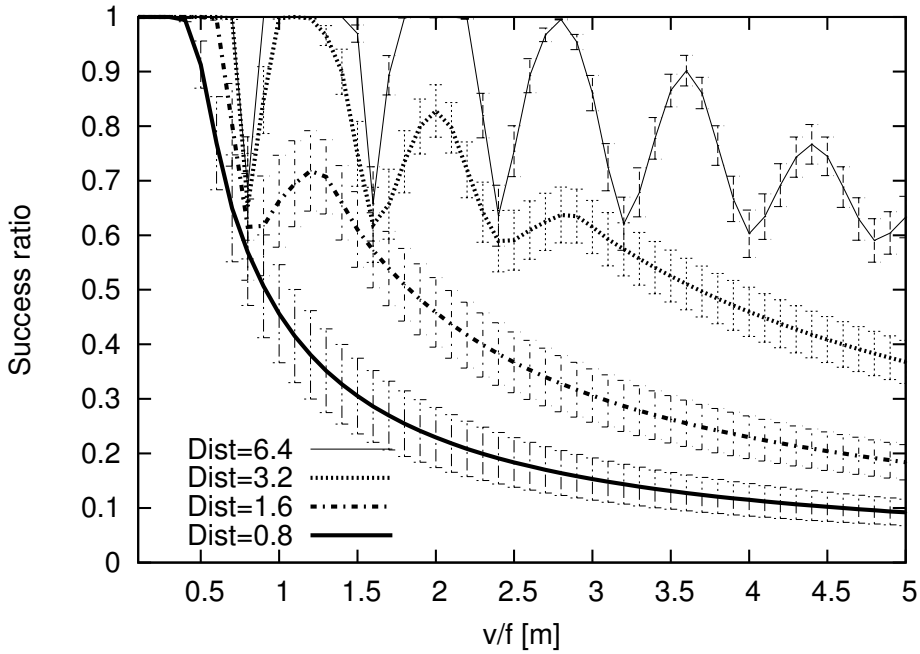

Figure 5.4: Probability of at least one successful authentication vs. inter-beacon distance $\frac{v}{f}$.

travel distance, which is consistent with the intuition that the success probability increases with the distance traveled: as the distance increases, the EV has more opportunities to broadcast at a position with a good packet success probability (e.g., right above the wireless device of the charging pad), and as a result the lines with larger total travel distance have more peaks (c.f. the concavity of the curves in Figure 5.3). Note that the peaks are 0.8 meters away from each other, which is due to that the charging pads are 0.4 meters long and are placed 0.4 meters away from each other. If the EV broadcasts every 0.8 meters, its relative position to its current charging pad will always be the same, and if the EV's first broadcast happens at a position with poor packet delivery ratio (e.g., at the edge of its current charging pad's communication range), its next broadcast will suffer from the same poor packet delivery ratio. Similarly, the first peak occurs at $\frac{v}{f} = 1.2$ meters because even if the EV's first broadcast is at the edge of the charging pad's range, its next broadcast 1.2 meters away will be right above the wireless device of the next charging pad.

We now compare the above model with the location-independent (LI) model proposed in [49], which assumes a location-independent packet success probability, and is an approximation to the above model. The LI model

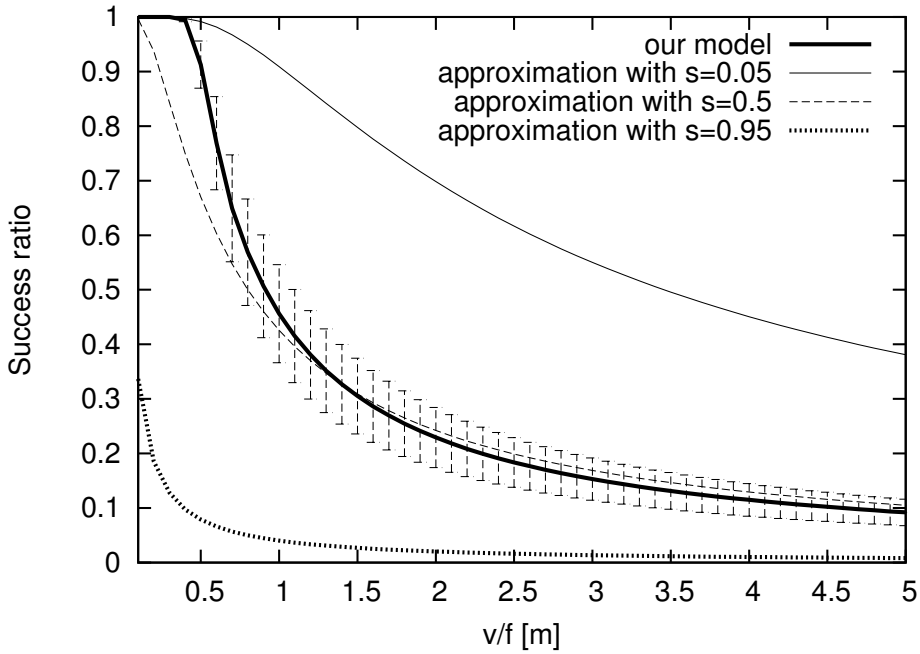

Figure 5.5: Comparison of our model with the approximate model in [49] for a total travel distance of 0.8 m and packet drop ratios $s = 0.05, 0.5, 0.95$.

considers that while traveling a distance of D, an EV will make a total of $D \cdot \frac{f}{v}$ broadcasts. Assuming that each beacon is dropped with probability s, the probability of at least one successful authentication is $1 - s^{D \cdot \frac{f}{v}}$.

In Figure 5.5 we compare the LI model from [49] with our model using a travel distance of $D = 0.8$ m (i.e., a single pad) for three values of the packet drop ratio ($s = 0.05, 0.5, 0.95$) The simple model with $s = 0.5$ approximates our model for larger values of $\frac{v}{f}$, but underestimates the success probability for lower values of $\frac{v}{f}$. This is because with lower values of $\frac{v}{f}$, the EV makes multiple broadcasts within the communication range of the charging pad, and our model considers the different success probabilities at the different broadcast positions.

5.4.3 Storage and Communication Overhead

To complete the assessment of the feasibility of Portunes, we quantify the approximate storage and communication requirements of the utility. We assume that the utility will store the EV's true identity, its assigned pseudonym and session key for each charging session, until the end of the current

monthly billing cycle, i.e., for up to 30 days. Assume that the EV's true identity e, pseudonym π, and the session keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ are all 128 bits long each. Then each entry takes $(128 * 4 =)512$ bits. According to current statistics there are about 250 million registered vehicles in the US, and assuming that all the vehicles are electric vehicles, total storage cost would be $(250 * 10^6 * 512 * 30 =)480$ GB, which is manageable given today's storage technology. Note that our calculation is an overestimation since the total storage cost is likely to be spread across multiple utilities in different areas of the nation.

Next, we quantify the communication overhead when the utility pre-distributes key material to the PO. The annual average daily traffic (AADT) of a highly congested road is generally in the order of hundreds of thousands of cars. Thus, we consider a congested road section with an AADT of 500,000 EVs. If the indexed key set is generated daily, then the utility needs to send at least 500,000 index-key tuples $(f(\pi), K^E_{f(\pi)}, K^A_{f(\pi)})$ to the PO. Assume we use SHA-1 as the one-way function f to derive the key index $f(\pi)$ from pseudonym π, and assume $K^E_{f(\pi)}, K^A_{f(\pi)}$ are both 128 bits, then each index-key pair costs $160 + 128 * 2 = 416$ bits, and the communication overhead incurred by daily key pre-distribution is $500,000 * 416$ bits $= 26$ MB, which can be easily delivered over a public network (from the utility to the PO). Furthermore, if the PO communicates with each charging pad using medium speed powerline communication (with typical data rates up to 576 kbits/s), the entire key set can be delivered from the PO to a charging pad in about 10 minutes, which shows that the communication requirements of Portunes can be met with off-the-shelf communication technologies.

5.5 Related Work

Authentication for dynamic charging can be viewed as a special case of Vehicle-to-Infrastructure (V2I) authentication, in that the infrastructure in question is a series of charging pads that (i) have very short range communication (several meters); and (ii) are placed closely to each other (tens of centimeters). These features distinguish dynamic charging authentication from authentication between vehicles and roadside units (RSUs) [25], and from authentication between EVs and static charging stations [50]. To the best of our knowledge, Portunes is the first work that focus on authentication in the dynamic charging scenario. Key pre-distribution based authentication was primarily used in wireless sensor networks [51], and has also been

adapted to vehicular network [52]. Our work differs in that EV e is authenticated using two keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ instead of a large subset of keys, and incurs less overhead during key transmission. One-time signatures [13, 14] 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 [12] 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 [30, 53] 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 [25, 54] 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 utility allocates pseudonym π and keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ to an EV before it enters a charging section. Portunes differs from existing works in that the keys $K^E_{f(\pi)}, K^A_{f(\pi)}$ are 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.

One popular option for privacy-preserving option is pseudonym-based schemes [55, 56, 57, 50, 58, 59], which allow the EV to authenticate with other entities using pseudonyms. Yang et al. [60] described a privacy-preserving protocol called P^2, where each EV uses a permit to authenticate itself with the connected power grid. The permit was generated by a trusted third party using partially blind signature and cannot be linked to the EV's real identity. Li et al. [55] proposed to use self-generated pseudonyms and identity-based signatures (IBS) to achieve privacy-preserving authentication. RAISE [56] is an RSU-based authentication mechanism that achieves k-anonymity. Nicanfar et al. [50, 58] considered the location privacy problem when the EV charges its battery at multiple charging stations, and suggested the use of different pseudonym at different charging stations. Mustafa et al. [61] considered a similar problem where the EV roams to another location and receives charging service from a different utility, where pseudonym is used to protect the EV's identity from the host utility.

Group signature [62] is another popular option for privacy-preserving authentication, and has been adopted in vehicular network [54, 63, 64, 65, 66]. In group signatures, a group leader generates keys for each member to sign

their messages. A verifier outside the group cannot infer the signer's identity from the signature, and only knows that the signer belongs to the group. In the vehicular network settings, one common approach is to use the RSU as group leader and treat all EVs within the RSU's communication range as its group members.

Researchers have also considered cooperative authentication [12, 67, 68, 69, 70, 71] for vehicular network. In cooperative authentication, each vehicle probabilistically verifies only a percentage of the messages, and cooperatively shares its verification result with other vehicles. In this way, cooperative authentication reduces the redundant verification of the same signature by different vehicles. The implicit authentication described in Section 5.2.4 follows the same intuition as cooperative authentication. The difference is that in our case, it is the charging pads that are cooperating with each other by sharing the location information of successfully authenticated EVs.

5.6 Conclusion

In this chapter, we presented Portunes, a privacy-preserving authentication protocol for EV to authenticate with wireless charging pads during dynamic charging. Portunes adopts a key-predistribution approach where the session keys are pre-distributed to the charging pads during idle period when there are less traffic on the road. This allows the protocol to bypass the key dissemination to charging pads in real time, and allows the EVs to perform lightweight authentication with the charging pads. By assigning unlinkable pseudonyms to the EV in different charging sections, Portunes also preserves the EV's location privacy. The implementation on Raspberry Pi indicates that message generation and verification using Portunes are both significantly 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.

CHAPTER 6

JANUS

In this chapter, we describe Janus, a privacy-preserving billing protocol for dynamic charging for the subscription-based billing model described in Section 2.4. In particular, in the bill calculation process, the utility is not able to learn when and where the EV has used dynamic charging. Our main idea is to embed homomorphic commitment of the price for each charging session as attributes in blind signatures signed by the utility. The EV and the PO compute their respective total fees locally and submit the values to the utility. The utility verifies that the total price is consistent with the combined homomorphic commitments. We implemented Janus in Python based on petlib [72] and evaluated the execution time on the Raspberry Pi platform. Our results show that all computations can be done within 0.6 seconds, which is well within the delay constraint for the subscription-based billing model.

The rest of this chapter is organized as follows: in Section 6.1, we introduce our models and key assumptions; in Section 6.2, we briefly describe security building blocks; in Section 6.3, we summarize key notations; in Section 6.4, we present the Janus protocol; in Section 6.5, we analyze security and privacy properties of Janus; in Section 6.6, we present performance evaluation results; in Section 6.7, we discuss several related issues; in Section 6.8, we review important related works; and we conclude this chapter in Section 6.9.

6.1 Model and Assumption

In this section we describe the models and assumptions.

6.1.1 Billing Model

Janus is designed for the subscription-based billing model introduced in Section 2.4. Recall that the billing model consists of two operations: fee

negotiation and fee aggregation.

- Fee negotiation happens prior to each dynamic charging session, where the EV and the PO negotiate and agree on the charging fee that the EV should pay for the coming dynamic charging session.
- Fee aggregation happens only once at the end of each billing cycle, where the EV calculates and submits to the utility its total fee that it should pay to the utility, and the PO calculates and submits to the utility the total fee that it should receive from the utility.

We refer the reader to Section 2.4 for a more detailed description of the billing model.

6.1.2 Communication Model

We assume the EV can communicate wirelessly with the utility and each PO through WiFi, DSRC, or cellular network. The particular communication technology is not of interest in this chapter. We also assume a fast reliable connection (e.g., Ethernet) between the utility and each PO.

6.1.3 Security Model

We assume that the utility is honest but curious, in that it faithfully follows the protocol but is interested to infer location information of individual EVs. We assume that in the fee negotiation phase of the subscription-based billing model, the EV and the PO are able to agree on the fee for each individual charging session. However, in the fee aggregation phase at the end of each billing cycle, we assume that the EV may attempt to underclaim its total fee that should be paid to the utility, and the PO may attempt to overclaim the total fee that it should receive from the utility.

6.1.4 Design Goals

Janus has two major design goals: correctness and privacy-preservation.

- Correctness: the correctness goal states that the total fees submitted by the EV and the PO to the utility in the fee aggregation phase of the billing model must be consistent with the charging fees of each individual charging session. In particular, the EV should be able to prove to the utility that it does not underclaim the total fee, and the

PO should be able to prove to the utility that it does not overclaim the total fee.

- Privacy-preservation: the privacy-preservation goal states that the protocol should minimize information available to the utility that can be used to infer location information of individual EVs. In particular, for each individual charging session, the time of the charging, the identity of the PO, and the charging fee for this individual charging session should all be hidden from the utility.

6.2 Security Building Blocks

In this section we describe the security building blocks used in the construction of Janus.

6.2.1 Homomorphic Commitment

A commitment scheme allows a user to bind a secret value x to a commitment C. The commitment C itself is information-hiding in that C does not reveal any information about x. Later the user can reveal the secret value x and prove that C is indeed a commitment of x. One example of perfect information-hiding commitment is the Pederson commitment scheme [73]: to commit a value x, the user chooses a random secret r and compute $C = g^x h^r$, where g and h are public. In order to prove that x is indeed committed in C, the user reveals x and r, and the verifier computes $C' = g^x h^r$ and checks that $C = C'$.

A homomorphic commitment scheme additionally allows one to obtain useful information by operating directly on commitments, without knowing the secret values in the commitments. A Pederson commitment can be extended to a homomorphic commitment as follows: given $C_1 = g^{x_1} h^{r_1}$ and $C_2 = g^{x_2} h^{r_2}$, we define

$$C_1 \boxplus C_2 = C_1 C_2 = g^{x_1+x_2} h^{r_1+r_2} \tag{6.1}$$

Note that anyone can compute $C_1 \boxplus C_2$, given only C_1 and C_2. Later the user can prove that $x_1 + x_2$ equals to the claimed value, without revealing x_1 and x_2 themselves, by revealing $r_1 + r_2$. In this chapter we use $Cmt(x_1, \ldots, x_n; r)$ to denote a Pedersen commitment with secret values $x_1, \ldots, x_n$ and commitment opener r.

6.2.2 Zero-Knowledge Proof

In Zero-Knowledge Proof (ZKP), the prover proves to the verifier possession of certain secret values that satisfy certain relations. The proof is zero-knowledge in the sense that the proof itself does not reveal any additional information about the secret values. We use notation $ZK\{x : P(x)\}$ to denote a zero-knowledge proof of secret value x such that x satisfies the relation $P(x)$. Every variable that appears to the left of the colon is a secret value only known to the prover, and every variable that only appears to the right of the colon is public. For example, the following notation $ZK\{x, r : g^x h^r = C\}$ represents a zero-knowledge proof that the prover knows the secret value x and the commitment opener r that is used to form the Pedersen commitment C.

A zero-knowledge proof is interactive if it involves real-time interaction (e.g., message exchanges) between the prover and the verifier. One example is the Schnorr Identification scheme [27], which, given public values g and h, allows the prover to prove knowledge of value x such that $g^x = h$ without revealing x. In Figure 6.1 we illustrate the Schnorr Identification scheme.

Prover (input=$x, g, h = g^x$)		**Verifier (input=g, h)**
$y \leftarrow$ random secret $a \leftarrow g^y$		
	$\xrightarrow{\quad a \quad}$	
		$c \leftarrow$ random secret
	$\xleftarrow{\quad c \quad}$	
$r \leftarrow y + cx$		
	$\xrightarrow{\quad r \quad}$	
		verify that $g^r = ah^c$

Figure 6.1: Schnorr Identification Scheme.

One can transform an interactive zero-knowledge proof into a non-interactive zero-knowledge proof (NIZKP) using the Fiat-Shamir Heuristic [74].

6.2.3 Blind Signature with Attributes

A blind signature scheme allows the user to obtain a signature from the signer while the signer does not learn the content of the message to be signed.

Baldimtsi and Lysyanskaya extended the definition of blind signature and introduced the concept of blind signature with attributes [27]. In addition to the message m, the user possesses certain secret attributes $L_1, \ldots, L_n$ and commits the attributes in a commitment $C = Cmt(L_1, \ldots, L_n; R)$ with commitment opener R. The commitment C is public while the message m and the attributes $L_1, \ldots, L_n$ are only known to the user. A blind signature with attribute would allow the user to obtain a signature σ on $(m, \tilde{C})$, where $\tilde{C}$ is a new commitment to the same attributes $L_1, \ldots, L_n$ but with a different opening secret $\tilde{R}$, i.e., $\tilde{C} = Cmt(L_1, \ldots, L_n; \tilde{R})$. The commitment opener $\tilde{R}$ of the new commitment $\tilde{C}$ is only known to the user. In this chapter we use the Anonymous Credential Light (ACL) [27] as the implementation of the blind signature with attributes scheme.

6.2.4 Single-Use Anonymous Credentials

An anonymous credential allows the user to prove possession of the credential without revealing the user's true identity. A single-use anonymous credential further guarantees that the credential can be used at most once. If the user attempts to spend a single-use credential more than once, the user's true identity can be revealed. In [27] the authors described a construction of single-use anonymous credentials using blind signatures with attributes. Let L_1 denote the user's true identity. To obtain a single-use anonymous credential, the user first generates a random secret L_0, and constructs a commitment $C = Cmt(L_0, L_1; R)$. The user then proves to the signer knowledge of L_0, L_1, R with respect to C and obtains a blind signature σ on $(m, \tilde{C})$, where m is a random message and $\tilde{C} = Cmt(L_0, L_1; \tilde{R})$. To spend the credential, the user reveals $m, \tilde{C}, \sigma$, receives a challenge c from the verifier, and then reveals the double-spending factor $d = cL_1 + L_0$. The verifier verifies that σ is a valid signature on $(m, \tilde{C})$. Note that if the user attempts to spend the same credential twice, the verifier would know $d = cL_1 + L_0$ and $d' = c'L_1 + L_0$, from which the user's true identity can be inferred as $L_1 = \frac{d-d'}{c-c'}$.

6.3 Notation

In this section, we summarize key notations used in the protocol. This section is meant for quick reference, and the reader should refer to Section 6.4 for detailed explanation of constructions such as the EV's anonymous cre-

dential and the receipts.

- H: one-way hash function.
- e: EV's true identity.
- p: PO's true identity.
- u: Utility's true identity.
- (e, p, u, i, j): index of the charging session in question, which is the i-th charging session of EV e, and the j-th charging session of PO p with any EV subscribed to utility u.
- $P = P_i^e = P_j^{p,u}$: the three variables all denote the same charging fee for a particular charging session agreed by EV e and PO p.
- $Cmt(L_0, L_1, \ldots, L_n; R)$: Pederson commitment of $(L_0, L_1, \ldots, L_n)$ with commitment opener R.
- $NIZK\{(x_1, \ldots, x_n) : P(x_1, \ldots, x_n, y_1, \ldots, y_m)\}$: Non-Interactive Zero-Knowledge proof of knowledge. The prover proves knowledge of secret values $x_1, \ldots, x_n$ which satisfy the relation $P(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where $y_1, \ldots, y_m$ are public values.
- $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$: single-use anonymous credential issued by the utility to EV e during the registration phase.
- sn_i^e: a random secret generated by EV e.
- $\hat{C}_i^e$: $\hat{C}_i^e = Cmt(L_i^e, e; \hat{R}_i^e)$ is a Pedersen commitment with random secret L_i^e and EV's identity e as committed values.
- L_i^e: random secret generated by EV e.
- c: challenge used in the double-spending equation $d_{i,j}^{e,p,u}$.
- $d_{i,j}^{e,p,u}$: $d_{i,j}^{e,p,u} = c \cdot e + L_i^e$ is the double-spending equation that allows identification of the EV's identity e if the same single-use credential τ_i^e is spent twice.
- $\hat{\sigma}_i^e$: utility's ACL signature on $(sn_i^e, \hat{C}_i^e)$.
- $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$: the receipt of EV e.
- σ_i^e: utility's ACL signature on $(m_i^e, \tilde{C}_i^e)$.

- m_i^e: $m_i^e = Cmt(e; z_i^e)$ is a Pedersen commitment with the EV's identity e as the committed value, and commitment opener z_i^e that is only known to EV e.
- $\tilde{C}_i^e$: $\tilde{C}_i^e = Cmt(P_i^e; \tilde{R}_i^e)$ is a Pedersen commitment with the charging fee P_i^e as the committed value, and commitment opener $\tilde{R}_i^e$ known to both PO p and EV e.
- $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$: the receipt of PO p.
- $\sigma_j^{p,u}$: utility's ACL signature on $(m_j^{p,u}, \tilde{C}_j^{p,u})$
- $m_j^{p,u}$: $m_j^{p,u} = Cmt(p; z_j^{p,u})$ is a Pedersen commitment with the PO's identity p as the committed value, and commitment opener $z_j^{p,u}$ that is only known to PO p.
- $\tilde{C}_j^{p,u}$: $\tilde{C}_j^{p,u} = Cmt(P_j^{p,u}; \tilde{R}_j^{p,u})$ is a Pedersen commitment with the charging fee $P_j^{p,u}$ as the committed value, and commitment opener $\tilde{R}_j^{p,u}$ only known to PO p.

6.4 Janus Protocol

In this section, we describe the Janus protocol in detail. Janus consists of three phases: registration, price validation, and reconciliation. In Figure 6.2, we illustrate when each phase is executed during the billing cycle.

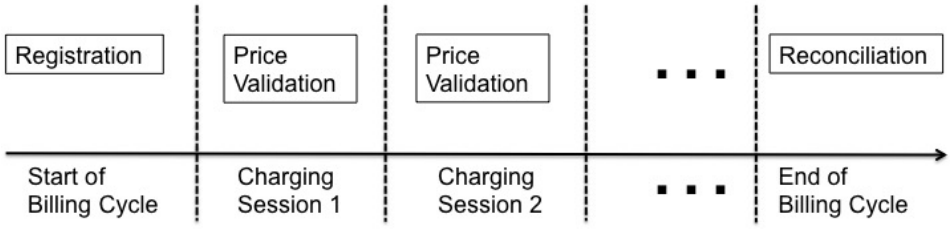

Figure 6.2: Illustration of Janus phases during the billing cycle.

- The registration phase happens once at the beginning of each billing cycle between each EV e and the utility, where the utility issues N single-use credentials $\tau_1^e, \ldots, \tau_N^e$ to EV e. For each dynamic charging session the EV must spend one unused credential.
- The price validation phase happens at the beginning of each dynamic charging session (after the EV and the PO have agreed on the charging

fee for the incoming charging session, and before the actual charging happens), where EV e, PO p and utility u run the price validation protocol described in Section 6.4.2. As a result, EV e obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and PO p obtains receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u, where σ_i^e is the utility's signature on the pair $(m_i^e, \tilde{C}_i^e)$, and $\sigma_j^{p,u}$ is the utility's signature on the pair $(m_j^{p,u}, \tilde{C}_j^{p,u})$. The price validation protocol uses blind signature so that the utility does not learn either $m_j^{p,u}$ nor $\tilde{C}_j^{p,u}$ during the signing process. The receipt proves that both the EV and the PO agreed on the price P, and the utility's signature prevents the EV or the PO from modifying the receipt.

- The reconciliation phase happens once at the end of the billing cycle, where EV e submits the total price $P^e = \sum_{i=1}^{M^e} P_i^e$ and all the receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ to the utility. PO p also submits the total price $P^{p,u} = \sum_{i=1}^{M^p} P_j^{p,u}$ and the validation tokens $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$. In addition each EV e also needs to reveal any unused credentials $\tau_{M^e+1}^e, \ldots, \tau_N^e$ to the utility.

We describe each phase in detail below.

6.4.1 Registration

The registration phase happens once at the beginning of each billing cycle. The main purpose of this phase is for the utility to issue anonymous credentials to the EV that will be used in the price validation phase. We assume that the EV authenticates with the utility using its true identity (e.g., its long-term public key) at the beginning of the registration phase, and the utility knows the EV's identity during the communication of the registration phase (but the utility does not learn the anonymous credentials issued to the EV until the EV spends it). We assume each EV is issued N credentials for each billing cycle. In Figure 6.3 we illustrate how EV e obtains a single-use credential $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$ from the utility as described in [27]. EV e obtains N credentials by repeating the protocol for N times. Below we describe the registration protocol:

- Step 1: to obtain the i-th credential, the EV first generates random secrets sn_i^e, L_i^e, and R. sn_i^e is a serial number that serves as the message to be signed, and L_i^e together with EV's true identity e serve as the attribute, as described in Section 6.2.3.

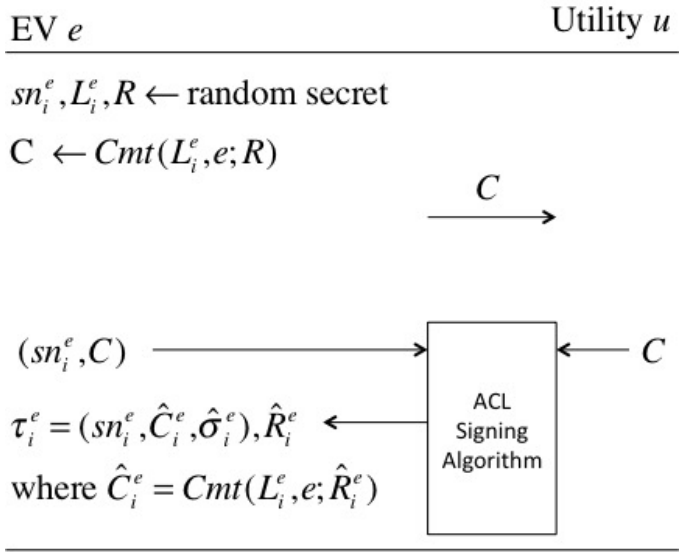

Figure 6.3: Overview of Registration phase of Janus. The above protocol shows how EV e obtains one single-use anonymous credential $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$ from the utility. To obtain a total of N credentials, the EV repeats the above protocol N times at the beginning of the billing cycle.

- Step 2: the EV commits L_i^e and its identity e in the commitment $C = Cmt(L_i^e, e; R)$ using secret R.
- Step 3: the EV runs the ACL signing protocol with the utility as the signer on the message sn_i^e and attribute commitment C. As a result, EV e obtains $\hat{C}_i^e, \hat{R}_i^e, \hat{\sigma}_i^e$, where $\hat{C}_i^e$ is a commitment to the same attributes (L_i^e, e) but with a different secret $\hat{R}_i^e$, i.e., $\hat{C}_i^e = Cmt(L_i^e, e; \hat{R}_i^e)$, and $\hat{\sigma}_i^e$ is the utility's signature on the pair $(sn_i^e, \hat{C}_i^e)$.

Note that during the signing process, the utility never learns the value of L_i^e, e or the new commitment $\hat{C}_i^e$, and the output signature $\hat{\sigma}_i^e$ cannot be linked to this signing session.

6.4.2 Price Validation

We observe that, when the EV uses dynamic charging service provided by some PO, the PO can physically observe the EV at its charging section, and thus there is no point hiding the EV's location information from the PO. A malicious PO can indeed disclose the identity of the observed EV to the utility or other entities. Such malicious PO behavior is out of our scope, and in this chapter we assume that the EV fully trusts the PO and the PO

does not disclose the EV's identity and location to any third party. Given this trust relationship between EV and PO, the PO can act as proxy and relay messages between the EV and the utility. In particular, the EV never communicates directly with the utility during the price validation phase.

In Figure 6.4 we illustrate the price validation protocol. We assume that this is the i-th charging session of EV e in the current billing cycle, and the j-th charging session of PO p with any EV subscribed to utility u. We thus denote this dynamic charging session by (e, p, u, i, j). We assume that EV e and PO p have agreed on the price P for this charging session. The EV records the price $P_i^e = P$ and the PO records the price $P_j^{p,u} = P$. The goal of the price validation phase is to let EV e obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and PO p obtains receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u. To preserve the EV's privacy, we use the ACL blind signature with attributes [27]. In particular, during the signing process, the utility does not learn the value of the price P, the EV's identity e, and cannot link the produced receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e), (m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ with the signing session.

The price validation protocol consists of 4 major steps: PO preparation, EV preparation, receipt generation, and EV validation. Below we describe each step in detail:

PO Preparation

- Step 1: PO receives a random nonce n generated by the utility.
- Step 2: PO generates a random secret $z_j^{p,u}$ and commits its identity p in $m_j^{p,u} = Cmt(p; z_j^{p,u})$.
- Step 3: PO generates two secrets $R_i^e, R_j^{p,u}$ and commits the price $P = P_i^e = P_j^{p,u}$ in the commitments $C_i^e = Cmt(P_i^e; R_i^e)$ and $C_j^{p,u} = Cmt(P_j^{p,u}; R_j^{p,u})$.
- Step 4: PO sends $C_i^e, R_i^e, C_j^{p,u}, R_j^{p,u}, n$ to the EV.

EV Preparation

- Step 1: EV verifies that the two commitments $C_i^e, C_j^{p,u}$ are formed correctly.
- Step 2: EV binds the nonce n generated by the utility with the two commitments into $c = H(n, C_i^e, C_j^{p,u})$ using a one-way function H, and

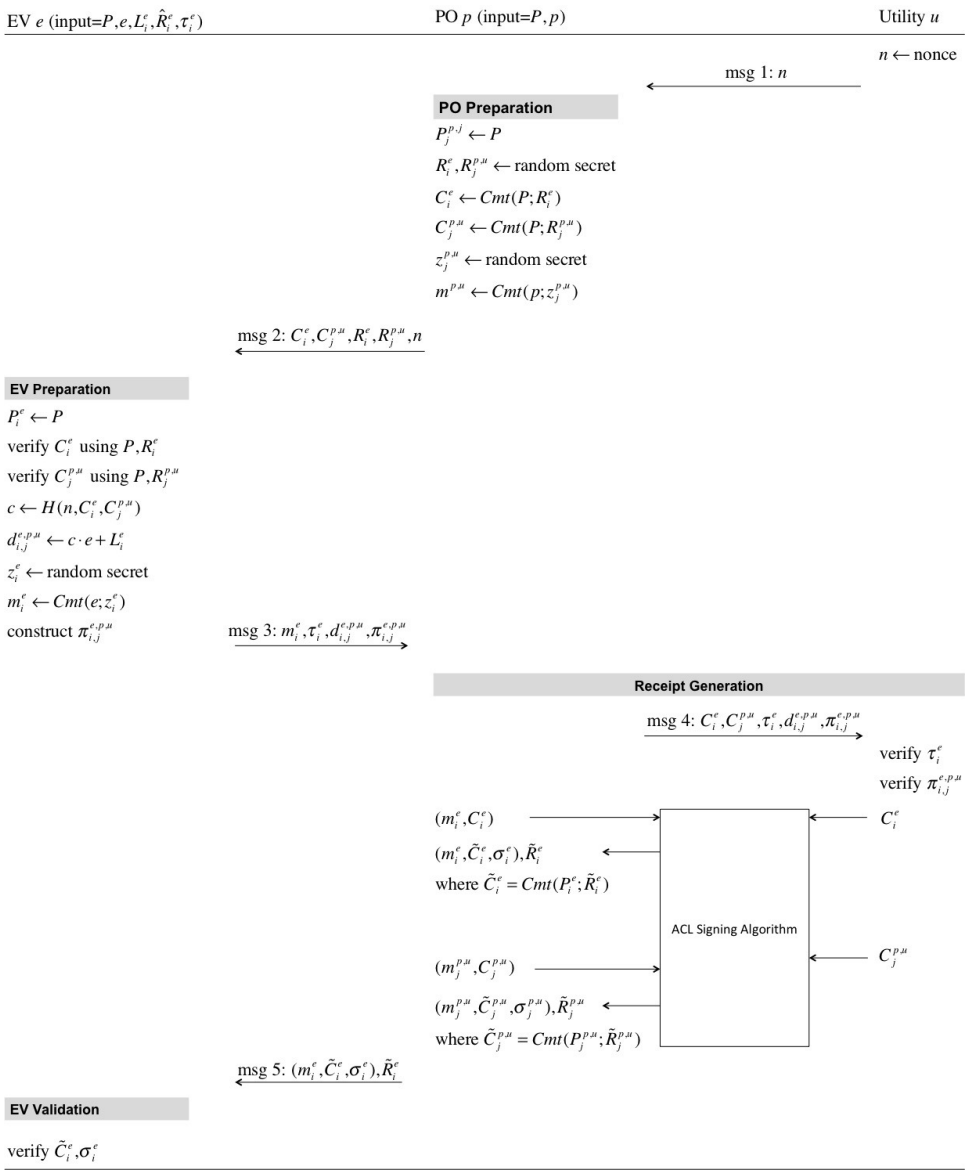

Figure 6.4: Overview of the Price Validation phase of Janus. We assume that this is the i-th time EV e receives dynamic charging service from any PO, and the j-th time that PO p provides dynamic charging service to any EV subscribing to utility u. As a result, EV e obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and PO p obtains receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u.

constructs the double-spending equation $d_{i,j}^{e,p,u} = c \cdot e + L_i^e$. The EV is essentially using c as the challenge to spend its single-use credential $\tau_i = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$.

- Step 3: EV then generates a random secret z_i^e and commits its identity e in $m_i^e = Cmt(e; z_i^e)$.

- Step 4: EV constructs a non-interactive zero-knowledge proof that (i) it knows the openings to C_i^e and $C_j^{p,u}$; and (ii) $d_{i,j}^{e,p,u}$ is correctly formed; and (iii) the single-use credential τ_i is correctly spent. The proof-of-knowledge equation is illustrated in equation 6.2.

$$\begin{aligned} \pi_{i,j}^{e,p,u} = NIZK\{(P_e, P_p, R_e, R_p, \hat{R}, e, L) : \\ C_i^e = Cmt(P_e; R_e) \wedge \\ C_j^{p,u} = Cmt(P_p; R_p) \wedge \\ \hat{C}_i^e = Cmt(L, e; \hat{R}) \wedge \\ d_{i,j}^{e,p,u} = H(n, C_i^e, C_j^{p,u}) \cdot e + L\} \end{aligned} \tag{6.2}$$

- Step 5: EV sends the $m_i^e, \tau_i^e, d_{i,j}^{e,p,u}, \pi_i^e$ to the PO.

Receipt Generation

- Step 1: PO sends to the utility $C_i^e, C_j^{p,u}, \tau_i^e, d_{i,j}^{e,p,u}, \pi_{i,j}^{e,p,u}$.

- Step 2: utility verifies that the credential τ_i^e is valid, and $\pi_{i,j}^{e,p,u}$ is correct.

- Step 3: utility runs the ACL signing algorithm as the signer with the PO on (m_i^e, C_i^e) and $(m_j^{p,u}, C_j^{p,u})$ respectively. In the end the PO obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, $\tilde{R}_i^e$, receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$, and $\tilde{R}_j^{p,u}$, where $\tilde{C}_i^e = Cmt(P_i^e; \tilde{R}_i^e)$ is a commitment to the same attributes P_i^e as C_i^e, but with a different secret $\tilde{R}_i^e$, and σ_i^e is the utility's ACL signature on $(m_i^e, \tilde{C}_i^e)$. Similarly, $\tilde{C}_j^{p,u} = Cmt(P_j^{p,u}; \tilde{R}_j^{p,u})$ is a commitment to the same attributes as $C_j^{p,u}$, and $\sigma_j^{p,u}$ is the utility's ACL signature on $(m_j^{p,u}, \tilde{C}_j^{p,u})$.

- Step 4: PO verifies that the receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ are formed correctly, and the commitment openers $\tilde{R}_i^e$, $\tilde{R}_j^{p,u}$ are valid.

- Step 5: PO stores its own receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ together with $\tilde{R}_j^{p,u}$

- Step 6: PO sends the other receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and the corresponding commitment opener $\tilde{R}_i^e$ to the EV.

Note that during the signing process, the utility does not learn the value of $m_i^e, \tilde{C}_i^e, m_j^{p,u}, \tilde{C}_j^{p,u}$.

EV Validation

In this step, the EV receives $(m_i^e, \tilde{C}_i^e, \sigma_i^e), \tilde{R}_i^e$ from the PO, and verifies that σ_i^e is indeed the utility's signature on $(m_i^e, \tilde{C}_i^e)$, and that $\tilde{R}_i^e$ opens the commitment $\tilde{C}_i^e = Cmt(m_i^e; \tilde{R}_i^e)$.

6.4.3 Reconciliation

The reconciliation phase happens at the end of the billing cycle. Each EV e computes the total sum that it should pay the utility u, and each PO p also computes the total sum that it should receive from utility u.

Assume that in the current billing cycle EV e has engaged in a total of M^e dynamic charging sessions (with any PO), and records the price P_i^e, the price receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, the secret $\tilde{R}_i^e$ that opens $\tilde{C}_i^e$, and z_i^e that opens m_i^e, for all $1 \leq i \leq M^e$. The EV constructs the total price

$$P^e = \sum_{i=1}^{M^e} P_i^e \tag{6.3}$$

and the commitment opener for the homomorphic commitment

$$R^e = \sum_{i=1}^{M^e} \tilde{R}_i^e \tag{6.4}$$

The EV then sends P^e, R^e together with the receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and z_i^e for all $1 \leq i \leq M^e$ to utility u. The utility checks that

- $Cmt(P^e; R^e) = \Pi_{i=1}^{M^e} \tilde{C}_i^e$
- $\forall i, m_i^e = Cmt(e; z_i^e)$
- σ_i^e is a valid ACL signature on $(m_i^e, \tilde{C}_i^e)$.

The EV also proves to the utility that it does not omit any payment, by revealing the rest $N - M^e$ unused credentials $\tau_{M^e+1}^e, \ldots, \tau_N^e$. If all the above verifications succeed, the utility accepts P^e as the correct total sum that EV e owns the utility for the billing cycle.

The reconciliation phase for the PO is almost identical. Assume that in the current billing cycle PO p has engaged in a total of M^p dynamic charging sessions with any EV subscribed to utility u. PO p construsts $P^{p,u} = \sum_{j=1}^{M^p} P_j^{p,u}$ and $R^{p,u} = \sum_{j=1}^{M^p} \tilde{R}_j^{p,u}$, and sends $P^{p,u}, R^{p,u}$ together with $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u}), z_j^{p,u}$ for all $1 \leq j \leq M^p$ to the utility. The utility checks that

- $Cmt(P^{p,u}; R^{p,u}) = \Pi_{j=1}^{M^p} \tilde{C}_j^{p,u}$
- $\forall i, m_j^{p,u} = Cmt(p; z_j^{p,u})$
- $\sigma_j^{p,u}$ is a valid ACL signature on $(m_j^{p,u}, \tilde{C}_j^{p,u})$.

If all verifications succeed, the utility accepts the value $P^{p,u}$ as the total sum it owes PO p. Since in the assumed billing model the utility should pay the PO, the PO has no economic incentive to omit a price value in computing the total sum $P^{p,u}$. Therefore, the protocol does not require the PO to prove to the utility that no price value $p_j^{p,u}$ is omitted.

6.5 Analysis

In this section we prove the correctness of Janus and analyze the location privacy it provides. Throughout the section we focus on EVs that contracted a particular utility u, and we denote by $\mathcal{E}$ the set of EVs that contracted the considered utility and by $\mathcal{P}$ the set of POs.

6.5.1 Correctness

To prove correctness, we have to show that the utility is able to verify that for each EV e the total charging fee P^e submitted by the EV is indeed the sum of the charging fees of all charging sessions that EV e participated in, i.e., $P^e = \sum_{i=1}^{M^e} P_i^e$. Similarly, we have to show that utility u is able to verify that the total charging fee $P^{p,u}$ claimed by PO p is not more than the sum of the charging fees of all charging sessions provided by the PO to EVs with a contract with u, i.e., $P^{p,u} \leq \sum_{j=1}^{M^{p,u}} P_j^{p,u}$.

First we show that the EV cannot omit payment.

In Janus, the utility does not compute the total fee owed by the EV, it only verifies that the total fee is consistent with the receipts, both of which are submitted by the EV. Naturally a malicious EV may attempt to underclaim the total fee by intentionally withholding submitting one or multiple receipts

$(m_i^e, \tilde{C}_i^e, \sigma_i^e)$. Janus mitigates this by requiring each EV to spend *all* of its single-use anonymous credentials τ_i^e. Recall that during the registration phase which happens once at the beginning of each billing cycle, the utility issues a total of N single-use credentials $\tau_1^e, \ldots, \tau_N^e$ to each EV e. In the price validation phase, in order to receive a receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, EV e must spend one of its unused credentials. Therefore, if during the reconciliation phase the EV submits a total of M receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e), 1 \leq i \leq M^e$, the EV must prove to the utility that it still possesses a total of $N - M^e$ unused credentials. The EV can prove this to the utility by repeating a simple challenge-response protocol for $N - M^e$ times: each time the EV receives a fresh challenge c from the utility it has to spend an unused credential by binding the credential to the value of c, in a way similar to how the EV binds the price commitments C_i^e and $C_j^{p,u}$ in the double-spending equation $d_{i,j}^{e,p,u} = H(n, C_i^e, C_j^{p,u}) \cdot e + L$ as we described in Section 6.4.2. If the EV fails to prove possession of $N - M^e$ unspent credentials, the utility knows that the EV is trying to omit payments, and can thus levy a fine on the EV. How high of a fine the utility levies is outside of the scope of the protocol.

Next we show that the EV and PO can only claim correct totals. Before we prove correctness of Janus in the sense formulated at the beginning of the section, we establish an important relationship between receipts obtained by an EV and a PO upon charging. For convenience, let us define the function Fee that extracts the charging fee from the commitment in a receipt, i.e.,

$$Fee((m, \tilde{C}, \sigma)) = P \text{ where } \tilde{C} = Cmt(P; \tilde{R}). \tag{6.5}$$

Consider the charging fees P_i^e committed in $\tilde{C}_i^e$, and the charging fees $P_j^{p,u}$ committed in $\tilde{C}_j^{p,u}$. There exists a bijective mapping f between the set $\{(m_i^e, \tilde{C}_i^e, \sigma_i^e) : \forall e \in \mathcal{E}, 1 \leq i \leq M^e\}$ and the set $\{(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u}) : \forall p \in \mathcal{P}, 1 \leq j \leq M^{p,u}\}$ such that $Fee((m_i^e, \tilde{C}_i^e, \sigma_i^e)) = Fee(f((m_i^e, \tilde{C}_i^e, \sigma_i^e)))$.

Consider an arbitrary receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ of an arbitrary EV e. Without loss of generality, assume EV e obtained this receipt in session (e, p, u, i, j). Observe that the price validation phase for charging session (e, p, u, i, j) starts with $P_i^e = P_j^{p,u}$. EV e then binds its credential τ_i to the two commitments $C_i^e = Cmt(P_i^e; R_i^e)$ and $C_j^{p,u} = Cmt(P_j^{p,u}; R_j^{p,u})$. Since PO j also knows the commitment opener R_i^e and $R_j^{p,u}$, it can verify that the EV indeed binds its credential to the two commitments C_i^e and $C_j^{p,u}$ instead of to some other commitments of different values. Therefore, when the PO relays the EV's zero-knowledge proof π to the utility, it is guaranteed that both PO p and EV e agree on the price $P_i^e = P_j^{p,u}$ committed in C_i^e and $C_j^{p,u}$. Since the

PO must relay the zero-knowledge proof π in order for the price validation protocol to complete, and since the EV and the PO can only obtain their receipts when the price validation completes, we are guaranteed that for the charging session (e, p, u, i, j) the price committed in $\tilde{C}_i^e$ of the EV's receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ must be equal to the price committed in $\tilde{C}_j^{p,u}$ of the PO's receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$. By defining $f((m_i^e, \tilde{C}_i^e, \sigma_i^e)) = (m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ for each charging session (e, p, u, i, j), we obtain the desired bijective mapping.

We can use the existence of a bijective mapping between receipts and commitments to prove the correctness of Janus. Janus is correct in the sense that it allows the utility to verify that $P^e = \sum_{i=1}^{M^e} P_i^e$ for each EV e and $P^{p,u} \leq \sum_{j=1}^{M^{p,u}} P_j^{p,u}$ for each PO $p \in \mathcal{P}$.

We give an indirect proof for the correctness of Janus, which it achieves using homomorphic commitments that are signed by the utility. Consider an EV e that during the reconciliation phase submits a total of $M^{e'}$ receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ where $1 \leq i \leq M^{e'}$. From the reconciliation phase it is clear that EV e can only claim a total fee that is equal to the sum of the charging fees committed in the commitments $\{\tilde{C}_i^e : 1 \leq i \leq M\}$. If the EV attempts to replace $\tilde{C}_i^e$ with another commitment $C' = Cmt(P'; R')$ with a different charging fee $P' \neq P_i^e$, it must forge the utility's signature σ' on (m_i^e, C'), which is infeasible given the security of the ACL signature [27]. Since $\tilde{C}_i^e$ is a commitment of P_i^e, this guarantees that the claimed total is $P^e = \sum_{i=1}^{M^{e'}} P_i^e$. Note that, so far there is no guarantee that the number of receipts $M^{e'}$ submitted by EV e is equal to the number of charging sessions M^e that the EV actually participated in. To guarantee that $M = M^e$, Janus requires the EV to reveal all unused credentials during the reconciliation phase. As discussed previously, this prevents EV e from intentionally omitting one or multiple individual charging fees in the calculation of the total fee. Given that $P^e = \sum_{i=1}^{M^{e'}} P_i^e$ and that $M^{e'} = M^e$, we have $P^e = \sum_{i=1}^{M^e} P_i^e$, which proves the first part.

The correctness of PO p's total fee follows a similar argument. PO p can only claim a total fee that is the sum of a subset of the charging fees committed in the commitments $\{\tilde{C}_j^{p,u} : 1 \leq j \leq M^{p,u}\}$. The PO cannot modify the value $P_j^{p,u}$ committed in $\tilde{C}_j^{p,u}$ without invalidating the signature $\sigma_j^{p,u}$ on the pair $(m_j^{p,u}, \tilde{C}_j^{p,u})$. The PO has to support its claimed total fee using a series of receipts $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$. Each receipt proves to the utility that some EV, whose true identity is unknown to the utility, has agreed on the fee that is committed in $\tilde{C}_j^{p,u}$. The utility can verify that the total fee claimed by the PO is indeed the sum of all of the fees committed in the homomorphic commitments $\tilde{C}_j^{p,u}$ that the PO submits, without learning the

value of the individual fees themselves, which proves the second part. PO p can thus claim at most an amount of $P^{p,u} = \sum_{j=1}^{M^{p,u}} P_j^{p,u}$.

Note that Janus does not prevent a PO from underclaiming the total fee, e.g., by intentionally omitting one or more receipts in the calculation of the total fee. Nonetheless, since in our billing model the PO does not receive payment directly from the EV but from the utility that aggregates the dynamic charging activities of the EVs during the past billing cycle, underlaiming the charging fee would only cause financial damage to the PO. A rational PO would thus not attempt to underclaim the total charging fee.

6.5.2 Location Privacy

To analyze the location privacy provided by Janus, recall that Janus allows the utility u to learn (P^e, M^e) about each contracted EV e, and $(P^{p,u}, M^{p,u})$ about each PO p, but it does not reveal to the utility the fee for each individual dynamic charging session, neither the charging sections that an EV has charged its battery at. If a utility has a single EV as customer then the aggregate information is clearly enough for the utility to invade the location privacy of the EV, i.e., to infer the charging sections the EV visited. In what follows we are interested in whether the utility can infer the set of charging sections that a particular EV visited, and possibly the fee for each possible charging session of an EV, in more likely scenarios.

For the analysis recall that $\sum_{e\in\mathcal{E}} M^e = \sum_{p\in\mathcal{P}} M^{p,u} = M$, and let us consider a particular outcome of charging sessions of EVs $e \in \mathcal{E}$ at POs $p \in \mathcal{P}$. We can model this by a bipartite multigraph $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \mathcal{S})$, where $\mathcal{S}$ is the set of edges, which contains an edge (e, p) for every charging session of EV e at PO p. Observe that $\mathcal{G}$ has parallel edges if any EV had multiple charging sessions at the same PO.

Without a priori information about the charging fees, inferring the EVs' location can be formulated as finding the (number of) bipartite multigraphs $(\mathcal{E}, \mathcal{P}, \mathcal{S})$ that satisfy

$$\sum_{e\in\mathcal{E}} \mathbf{1}_{\mathcal{S}}((e,p)) = M^{p,u} \quad \forall p \in \mathcal{P} \tag{6.6}$$

$$\sum_{p\in\mathcal{P}} \mathbf{1}_{\mathcal{S}}((e,p)) = M^{e} \quad \forall e \in \mathcal{E}, \tag{6.7}$$

and that allow a feasible vector of payments $(P_{(e,p)})$ to the problem

$$\sum_{p\in\mathcal{P}} P_{(e,p)} = P^e, \quad \forall e \in \mathcal{E} \tag{6.8}$$

$$\sum_{(e,p)\in\mathcal{S}} P_{(e,p)} = P^{p,u}. \tag{6.9}$$

Constraint (6.6) corresponds to the number of charging sessions of PO p, (6.7) to the number of charging sessions of EV e, (6.8) ensures that the total fee of each EV is allocated, and (6.8) enforces a feasible allocation of fees between EVs and POs.

In the worst case every bipartite multigraph that satisfies (6.6) and (6.7) allows a feasible vector of payments. The following result shows that if each PO provides charging sessions to sufficiently many EVs then the number of feasible bipartite graphs grows exponentially.

The number of bipartite graphs that satisfy (6.6)-(6.7) is lower bounded by

$$\left(\frac{M}{|\mathcal{E}||\mathcal{P}|}\right)^{\Omega(|\mathcal{E}||\mathcal{P}|)}. \tag{6.10}$$

To obtain the number of bipartite multigraphs satisfying the above constraints, let us consider the biadjacency matrix of a bipartite multigraph, i.e., the non-negative integer valued matrix of size $|\mathcal{E}|\times|\mathcal{P}|$ whose entry (e,p) is the number of parallel edges between vertices e and p. Every bipartite multigraph satisfying (6.6) and (6.7) has a biadjacency matrix whose row sum for row e is M^e and column sum for column p is $M^{p,u}$. Finding the feasible bipartite multigraphs is thus equivalent to finding the non-negative integer valued matrices of size $|\mathcal{E}|\times|\mathcal{P}|$ with row sum sequence $(M^e)_{e\in\mathcal{E}}$ and column sum sequence $(M^{p,u})_{p\in\mathcal{P}}$. While it is known that a feasible matrix can be found in polynomial time, counting the number of feasible matrices is known to be #P-hard even for $|\mathcal{E}| = 2$ [75]. Furthermore, if the matrix is dense, i.e., $min_{e\in\mathcal{E}}\{M^e\} = \Omega(|\mathcal{P}|)$ and $min_{p\in\mathcal{P}}\{M^{p,u}\} = \Omega(|\mathcal{E}|)$, then the number of bipartite graphs can be lower bounded by [76]

$$(\frac{M}{|\mathcal{E}||\mathcal{P}|})^{\Omega(|\mathcal{E}||\mathcal{P}|)}. \tag{6.11}$$

The above result shows that the search space grows exponentially with both the number of EVs and the number of POs. For example, suppose there are a total of 10^4 EVs and 10 POs, and each EV participates in 2 charging sessions on average per day. Suppose the billing cycle is 30 days, then there will be a total of $M = 6 * 10^5$ charging sessions in the billing cycle. To

consider all feasible bipartite graphs, i.e., which EV visited which PO for how many times, the utility needs to search in a space of $(\frac{6*10^5}{10^4*10})^{\Omega(10^4*10)} \sim 6^{10^5}$. Thus, without a priori information about the charging fees Janus provides a high level of location privacy when there are sufficiently many EVs and POs.

To evaluate the case when the utility does have a priori information we now consider the case when a PO charges the same amount for every charging session. To consider this case it suffices to introduce the additional constraint

$$P_{(e,p)} = P^{p,u}/M^{p,u} \quad \forall (e,p) \in \mathcal{S}, \tag{6.12}$$

i.e., the fee of a charging session at PO p is always $P^{p,u}/M^{p,u}$.

Next we show that the problem of finding a bipartite graph and a feasible vector of payments that satisfy (6.6)-(6.9) and (6.12) is NP-hard.

It is easy to see that the problem defined by (6.6)-(6.9) and (6.12) corresponds to the sized multiple subset sum problem, which is a generalization of the sized subset sum problem [77]. The sized subset sum problem consists of a list of positive integers (the charging fees of the POs, one integer per charging session), a positive integer P^e, and a positive integer parameter M^e. The objective is to decide whethere there is a sublist of size M^e of the integers that sums to P^e. The sized subset sum problem is W[1]-hard [77], i.e., its complexity increases exponentially in the parameter M^e, and is exactly the problem of assigning charging sessions that satisfy (6.8) to an EV e. Since the problem has to be solved for all EVs simultaneously, the problem (6.6)-(6.9) and (6.12) is a generalization of the sized subset sum problem and is thus NP-hard.

To summarize, without a priori information it is the number of feasible bipartite graphs that makes privacy invasion infeasible, while with a priori information it is the computational complexity. An analysis of the complexity of identifying charging sessions under different priors is a topic on its own right, and is beyond the scope of this chapter.

6.6 Evaluation

In this seciton we present the evaluation result of Janus.

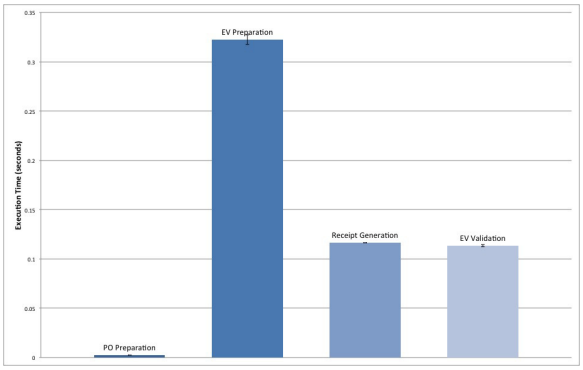

Figure 6.5: Execution time of the price validation protocol. EV preparation and EV validation are run on a Raspberry Pi 2 Model B. PO preparation and signature generation are run on a Macbook. Error bars indicate 95% confidence intervals.

6.6.1 Implementation

We have implemented Janus in Python using the petlib library [72], which includes an implementation of the ACL signature scheme. The implementation uses Pederson commitment as the homomorphic commitment scheme and P-224 elliptic curve group. Since the implementation is primarily meant to evaluate the execution time rather than the communication delay, we implemented the protocols as a single process and simplified message passing between different entities as function calls.

6.6.2 Execution Time

The Janus protocol consists of 3 phases, but the registration and the reconciliation phases are only executed once per billing cycle, at the beginning and at the end, and rely only on ACL signatures and homomorphic commitments. The most complex part of the Janus protocol is the price validation phase shown in Figure 6.4, which is executed at the beginning of each dynamic charging session, and thus we focus on the execution time of this phase. For the evaluation we consider a reasonable scenario in which the PO and the utility have more computational power than the EV, and thus we run the EV preparation and the EV validation steps on a Raspberry Pi 2 Model B, which has a 900MHz quad-core ARM Cortex-A7 CPU and 1GB RAM, and run the PO preparation and the receipt generation steps on a macbook pro with a 2.7GHz Intel Core i5 and 8 GB RAM. We repeat the execution for 20 times, and we show the average execution time of the various steps in Figure 6.5.

Clearly, the most time-consuming operations are included in the EV preparation step, where the EV constructs the non-interactive zero-knowledge proof $\pi_{i,j}^{e,p,u}$. The signature generation step, although involving more cryptographic operations, takes less time to execute since the utility and the PO have more computational power than the EV. Overall the price validation protocol took less than 0.6 seconds to execute, which makes it practical for the subscription-based billing model, even if communication delays are considered.

6.6.3 Communication Overhead

Recall that Janus consists of three phases: registration, price validation, and reconciliation. Both the registration and the reconciliation phase happen only once per billing cycle, and the delay constraint on these two phases are very loose. We therefore are more interested in the communication overhead of the price validation phase, which happens once per charging session.

The price validation phase involves two instances of ACL signature generation, each of which involves 5 message exchanges [27]. Nonetheless, the ACL signature generation algorithm is run by the PO and the utility, hence the time needed for 5 message exchanges is not significant. We thus treat ACL signature generation as a blackbox, and refer to [27] for its analysis.

Besides the message exchanges used in the ACL algorithm, the price validation phase requires only 5 messages. In Table 6.1, we show the sizes of the messages exchanged in Figure 6.4[1]. The numbers correspond to the sizes of the actual python objects in our implementation. The two largest messages are msg 3 and msg 4, which include the non-interactive zero-knowledge proof $\pi_{i,j}^{e,p,u}$ and the EV's credential τ_i^e. Recall that the price validation phase during which these message exchanges occur happens once per charging session, their transmission over any reasonable wireless communication system would incur a small transmission time. These experimental results show that the computational and communication complexity of Janus make it practical for dynamic EV charging.

6.6.4 Scalability

In Janus, the EV obtains all its single-use anonymous credentials τ_i^e from the utility during the registration phase. Issuing one anonymous credential requires running the signing algorithm of ACL signature, which is time

[1]We omitted the message exchanges in the standard ACL signature signing process.

Message	Size (bytes)
msg 1	28
msg 2	180
msg 3	1117
msg 4	1165
msg 5	833

Table 6.1: Message sizes

consuming (e.g., taking around 1 sec to complete). However, this does not affect the scalability of the protocol because the registration phase happens once at the beginning of the billing cycle, where the EV has several hours or even days to obtain all the anonymous credentials it needs. Similarly, the reconciliation phase also happens once every billing cycle, and does not affect the real-time scalability of Janus.

6.7 Discussion

To put Janus into a context, we continue with a discussion of topics related to the design of Janus.

6.7.1 Comparison with Electronic Toll Pricing

If we regard each dynamic charging section as a toll road, then electronic toll pricing protocols [78, 79, 80] can be used in dynamic charging. However, to the best of our knowledge, most electronic toll pricing protocols require random spot checks to combat the malicious behavior where the vehicle drives through the toll road without running the protocol, e.g., by switching off the vehicle's on-board communication device. Random spot check incur additional maintenance cost, e.g., deployment of patrol cars, and may also raise fairness issues in certain cases, e.g., short-term rental car service. Janus does not rely on random spot check at all to detect payment omission. One important difference between the scenario of dynamic charging and that of electronic toll pricing is that, in electronic toll pricing, even if the vehicle does not own the proper authorization and does not authenticate itself, it can still drive on the toll road segment (unless there is a physical gate enforcing proper payment before entry). To combat driving on toll roads without proper authentication and payment, plate-reading cameras could capture the plate number of violating vehicles and the driver would be responsible for paying a fine. However, in the dynamic charging scenario,

authentication between the EV and the charging pads must complete before the EV's battery can be charged. If the EV chooses to turn off its communication device and does not authenticate with the charging pads, but simply drives through the dynamic charging section, it is not a violation because the charging pads will simply not charge the EV's battery, and there is no loss for either the pad owners or the utility. Janus utilizes the fact that the EV must authenticate itself before battery charging, and effectively binds authentication with payment: the EV must first prove to the pad owner its authorization by spending the single-use anonymous credential obtained from the utility. The anonymous credentials serve as both an authentication token and a binding of the dynamic charging fee to the payment token.

6.7.2 Comparison with Direct Billing Model using Digital Cash

An alternative billing model for dynamic charging might be for the EV to directly pay the PO using digital cash for each dynamic charging session. Anonymous digital cash such as Zerocoin [81] or Zerocash [82] can be used to implement the financial transaction between the EV and the PO. One might argue that, given the possibility of digital cash, it is not necessary to have billing protocol for the subscription-based billing model.

One drawback of using digital cash under the direct billing model is that the EV must pre-load funds into its account and make sure that the account has sufficient balance before entering the charging section, and the EV may thus run out of funds at inconvenient times. This problem does not exist in the subscription-based billing model that Janus is designed for, where the EV does not need to pre-load funds, and makes the actual payment only at the end of the billing cycle.

Note that the subscription-based billing model and the direct billing model described above are not mutually exclusive. Just like a store may accept both credit card and cash payment, a PO may accept both direct payment using digital cash and indirect payment using Janus. It is thus important to clarify the distinct advantages provided by the subscription-based billing model. The subscription-based billing model, however, enables flexible pricing plans that are not provided by the direct billing model, e.g., the EV can purchase a plan of 1000 miles from the utility and use dynamic charging anywhere anytime to recharge its battery up to 1000 miles of total driving distance.

We also note that Janus is designed specifically for dynamic charging scenarios, whereas digital cash is designed for more general scenarios, and thus there are certain features that a digital cash scheme can provide but

Janus cannot. For example, a digital coin can be transferred multiple times among different entities, whereas in Janus the EV can only spend a credential τ once, and can only spend it with the utility. Nonetheless, because of the generality that digital cash aims to provide, it generally involves more complex designs than Janus. Anonymous digital cash designs may require even more complex cryptographic operations or zero-knowledge proofs to guarantee anonymity during spending. This may result in longer execution time of digital cash operations. For example, the Zerocash [82] design involves a zero-knowledge succinct non-interactive argument of knowledge (zk-SNARKs) that takes more than 2 minutes to generate on a platform with Intel Core i7-2620M 2.7GHz CPU and 12GB RAM. In comparison, the zero-knowledge proof π used by Janus takes less than 0.4 seconds to generate on the portable Raspberry Pi 2 platform with significantly less computational power (900MHz CPU and 1GB RAM).

6.7.3 Trust relationship between EV, Pad Owner, and the Utility

We finally discuss our assumption concerning the trust relationship between the EV, the PO, and the utility. Our billing model is a subscription-based model, in that the EV subscribes to the utility and receives a monthly bill aggregating all its dynamic charging usage from the utility. This requires the EV to trust the utility to make the correct calculation. In Janus, since the EV knows the ground truth of the dynamic charging fee for each charging session, it can easily verify if the total price in the monthly bill is correct. The utility can verify that the EV does not omit any payment token, and each payment token is authorized by some PO. However, the trust relationship between the EV and the utility does not automatically imply that the EV should give up its location privacy to the utility. Janus protects the EV's location privacy by not allowing the utility to infer the fee of a particular individual charging session from the payment tokens submitted by the EV.

To a certain extent, Janus assumes more trust relationship between the EV and the PO. This is reflected in the design where the PO effectively acts as a proxy between the EV and the utility: the EV does not directly communicate with the utility during the price validation phase; instead, the PO relays the credentials of the EV to the utility and the payment token from the utility to the EV. We justify this design choice by the observation that the PO is able to physically observe the EV. By our definition, the PO operates the dynamic charging section, and if the EV drives over the dynamic charging section, the corresponding PO inevitably observes the

EV. We thus argue that the billing protocol, which works within the cyber space, cannot prevent the PO from revealing the EV's location and identity to any other entity in the physical space, and the billing protocol must assume that the PO will not reveal the EV's location information. As a consequence, mechanisms are needed to discourage a PO from revealing the EV's identity and location to a third party, whether or not Janus is used. The design of such mechanisms is outside of the scope of the chapter.

6.7.4 Charging Fee Negotiation

Janus assumes that the EV and the PO are able to negotiate charging fees for each dynamic charging session. The problem of charging fee negotiation is orthogonal to the problem space of Janus, whose major goal is to allow verifiable aggregation of per-session charging fee into total charging fee without compromising the EV's location privacy. The EV and the PO may negotiate charging fee for the incoming dynamic charging session according to various pricing policies, e.g., fixed-rate pricing, day-ahead pricing, real-time pricing, etc., and the exact policy and negotiation protocol used by the EV and the PO is not our concern in this chapter.

Janus requires the EV and the PO to negotiate charging fee and complete the price validation phase *prior* to the charging. Note that once the price validation phase completes, the PO already obtains the receipt that it can use to claim money from the utility. In particular, a malicious PO may complete price validation with the EV but refuses to charge the EV. One might suggest that the PO should first charge the EV's battery, and only after the dynamic charging finishes do they run the price validation phase. However, this alternative design choice would allow a malicious EV to freeride the dynamic charging service by simply not running the price validation phase after it has received electricity from the PO. The question of whether price validation should complete before or after the actual charging is thus related to the question whether the EV or the PO is more likely to behave maliciously. We made the design choice where the EV and the PO complete price validation before the actual charging for the following reasons: (i) this is consistent with our assumption that the EV fully trusts the PO; (ii) a malicious PO is more likely to be caught, since the dynamic charging section is a physical road segment that does not move, and if the PO is reported of behaving maliciously, e.g., not charging the EV's battery after price validation completes, the utility or some other authority could send their own EVs to collect evidence; and (iii) in the scenario where multiple

POs co-exist in the area and compete with each other, a PO not honoring the negotiated charging amount is likely to lose customers.

Another advantage of having the EV complete price validation phase with the PO prior to charging is authentication. Recall that in the price validation phase, EV e must spend a single-use anonymous credential τ_i^e. The PO is able to verify that the credential is spent correctly, which in turn tells the PO that this EV is valid and indeed subscribes to the utility. If the EV fails to spend an unused anonymous credential, the PO considers the EV as unauthenticated and will simply not charge the EV's battery at all.

6.8 Related Work

Several works have been proposed to use modern cryptography to improve privacy of electronic toll pricing service [78, 79, 80]. One common feature shared by these designs is that the vehicle is required to periodically broadcast certain information that can be used later by the authority to calculate its bill. The challenge is to make sure that the information disclosed during the periodic broadcast and the bill calculation process do not violate the vehicle's location privacy. In VPriv [78] the vehicle periodically broadcasts tags whose commitment the vehicle has registered with the authority. To calculate the bill, the authority sends to the vehicle the prices associated with each tag that the authority has received from any vehicle, and the vehicle calculates the total price using the prices corresponding to its own tags. The authority and the vehicle then engage in a two-party protocol where the vehicle proves either of the following: i) the tags used in the calculation are valid; or ii) the total price is calculated correctly with respect to the tags. The two-party protocol can be executed multiple times to improve the authority's confidence that the vehicle's bill is calculated correctly. Unlike VPriv, PrETP [79] does not use two-party computations. Instead, the vehicle periodically broadcasts a homomorphic commitment of the price corresponding to the current road segment. During the bill calculation phase, the authority combines all the individual commitment of the vehicle to obtian a commitment of the total fee; the vehicle calculates the total fee and proves to the authority that it knows the opening to the commitment of the total fee. To guarantee that an EV would faithfully broadcast the information required by the protocol, both VPriv and PrETP rely on random spot checks, and if the vehicle is physically observed at certain time and location, the vehicle is responsible for providing the proof that

the toll price corresponding to that time and location is included in the total fee correctly. Milo [80] improves PrETP by considering the possibility that multiple vehicles collude and share the location of random spot checks with each other, and uses blind identity-based encryption to guarantee that the vehicles would not be able to learn these random spot check locations. In Spectre [83], the vehicle is given certain amount of Chaum's e-cash [84] as tokens, and periodically broadcast tokens while driving (each token can be spent only once). At the end of the billing cycle, the vehicle submits all unused tokens, and pay for the tokens spent on the road. The drawback shared by VPriv, PrETP, Milo, and Spectre is that they all rely on random spot checks to guarantee that the vehicle is following the protocol faithfully. The random spot check incurs additional cost for the authority, and to some degree violates the vehicle's location privacy as well.

Privacy-preserving billing has also been proposed for other transportation scenarios. Liu et al. [85] proposed a privacy-preserving billing scheme with revocation for static charging of electric vehicles based on the BBS+ signature [86]. The static charging scenario, where an EV stops at a charging station to charge its battery, is quite different from dynamic charging where an EV charges its battery while moving on the road. Kerschbaum et al. [87] proposed a privacy-preserving billing mechanism for public transportation (e.g., subway or bus). The scenario in [87] assumes that the user must tap a special cash card at the gate before entering or exiting the transportation system, and that the user will add value to the cash card at a special machine from time to time.

6.9 Conclusion

In this chapter we have presented Janus, a privacy-preserving billing protocol for dynamic charging of electric vehicles. By using blind signatures with attributes and homomorphic commitments, Janus allows the utility to verify that the total payment of the individual EVs and the total fee that the PO should receive are calculated correctly, without learning the dynamic charging fee of each individual charging session. Our python-based implementation indicates that the most real-time price validation phase of Janus can complete within 0.6 seconds.

CHAPTER 7

CONCLUSION

Dynamic charging is a promising technology for future electrified transportation. By allowing electric vehicles (EVs) to charge their batteries while moving, dynamic charging potentially increases the driving range of EVs and reduces the battery size and in turn the total price of the EV. However, dynamic charging is also sensitive to the variation of physical parameters such as the EV's speed, alignment, and the airgap between the charging pad and the EV, etc. Cyber infrastructure support is a pre-requisite in order for the charging pads to learn the necessary charging parameters before charging the incoming EV's battery. Since the messages exchanged between the EV and the charging facilities may contain sensitive information such as the EV's current battery State-of-Charge (which can be used to infer the EV's past locations), the cyber infrastructure must provide secure and privacy-preserving authentication and communication for dynamic charging. Towards this end we have presented two protocols: FADEC and Portunes. FADEC is a general V2I authentication protocol that allows EVs to efficiently authenticate with a series of roadside units (RSUs). FADEC adopts a proactive key dissemination approach to achieve seamless authentication handoff, and allows the EV to authenticate with a sequence of RSUs without re-negotiating the session key. Portunes is an authentication protocol specifically for the dynamic charging scenario, where the EV must authenticate with charging pads at high frequency (e.g., once every 30 milliseconds). Portunes adopts a key pre-distribution approach, where the computationally intensive parts of the protocol such as key generation and distribution are performed during the night when most EVs are parked. This allow EVs to authenticate with charging pads using only lightweight symmetric cryptographic operations. We described Janus, a privacy-preserving billing protocol for the subscription-based billing model, where the EV receives a single bill from the utility that aggregates all its dynamic charging activities in the past billing cycle. Janus takes advantage of modern cryptographic building blocks such as blind signature with attributes, and allows the util-

ity to verify the correctness of the EV's bill without learning the time and location of each dynamic charging session of the EV, which preserves the EV's location privacy. Our evaluation results based on simulations and implementations indicate that FADEC, Portunes, and Janus are efficient and practical for future dynamic charging applications.

REFERENCES

[1] H. Perik, "Practical EV Integration Cases for Static and Dynamic Wireless Power Transfer," in *IETEVC*, 2013.

[2] G. Covic and J. Boys, "Modern Trends in Inductive Power Transfer for Transportation Applications," *IEEE ESTPE*, 2013.

[3] 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*.

[4] 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*.

[5] "Vehicle Technologies Office Merit Review 2015: Wireless Charging of Electric Vehicles." [Online]. Available: http://energy.gov/eere/vehicles/downloads/vehicle-technologies-office-merit-review-2015-wireless-charging-electric

[6] "Federal Transit Administration Report No. 0060: Review and Evaluation of Wireless Power Transfer (WPT) for Electric Transit Applications." [Online]. Available: http://www.fta.dot.gov/documents/FTA_Report_No._0060.pdf

[7] "UK to trial in-road wireless charging tech for electric vehicles." [Online]. Available: http://www.gizmag.com/uk-electric-highways-trial/38897/

[8] [Online]. Available: http://www.electric-vehiclenews.com/2015/08/uk-to-test-dynamic-wireless-charging.html

[9] "SAE J1772: Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler."

[10] "Plugless Wireless Charging System." [Online]. Available: https://www.pluglesspower.com/

[11] [Online]. Available: http://www.electric-vehiclenews.com/2015/08/uk-to-test-dynamic-wireless-charging.html

[12] 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.

[13] L. Reyzin and N. Reyzin, "Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying," in *ACISP*, 2002.

[14] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *IEEE SP*, 2000.

[15] E. B. Barker and A. L. Roginsky, "SP 800-131A. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths," Gaithersburg, MD, United States, Tech. Rep., 2011.

[16] W. Diffie, P. Van Oorschot, and M. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[17] A. C.-C. Yao and Y. Zhao, "OAKE: A New Family of Implicitly Authenticated Diffie-hellman Protocols," in *ACM CCS '13*.

[18] W. Aiello, S. M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A. D. Keromytis, and O. Reingold, "Just fast keying: Key agreement in a hostile internet," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 2, May 2004.

[19] I. Marsh, "VANET communication: A traffic flow approach," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, Sept 2012, pp. 1043–1048.

[20] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 16–22, 2009.

[21] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO - Simulation of Urban MObility: An Overview," in *SIMUL*, 2011.

[22] P. VARAIYA, "What Weve Learned About Highway Congestion," *Access*, vol. 27, 2005.

[23] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, 2011.

[24] "IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006)."

[25] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs," in *SECON '09*.

[26] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to Win the Clonewars: Efficient Periodic N-times Anonymous Authentication," in *ACM CCS '06*.

[27] F. Baldimtsi and A. Lysyanskaya, "Anonymous Credentials Light," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 1087–1098.

[28] J. Camenisch and A. Lysyanskaya, *Advances in Cryptology – CRYPTO 2004: 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ch. Signature Schemes and Anonymous Credentials from Bilinear Maps, pp. 56–72.

[29] H. Li, G. Dán, and K. Nahrstedt, "Lynx: Authenticated Anonymous Real-Time Reporting of Electric Vehicle Information," in *IEEE International Conference on Smart Grid Communications (SmartGridComm), 2015*.

[30] "RFC 5201: Host Identity Protocol."

[31] J. So and J. Wang, "Micro-HIP A HIP-Based Micro-Mobility Solution," in *ICC Workshops*, 2008.

[32] N. Saxena and N. Chaudhari, "An Efficient Batch Verification Protocol for Value Added Services," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 1560–1565.

[33] S. Jiang, X. Zhu, and L. Wang, "A conditional privacy scheme based on anonymized batch authentication in Vehicular Ad Hoc Networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, April 2013, pp. 2375–2380.

[34] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[35] H. Guo, Y. Wu, F. Bao, H. Chen, and M. Ma, "UBAPV2G: A Unique Batch Authentication Protocol for Vehicle-to-Grid Communications," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 707–714, Dec 2011.

[36] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. Khan, "b-SPECS+: Batch Verification for Secure Pseudonymous Authentication in VANET," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 11, pp. 1860–1875, Nov 2013.

[37] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in Vehicular Ad Hoc Networks," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 1, pp. 248–262, Jan 2011.

[38] H.-R. Tseng, "On the security of a unique batch authentication protocol for vehicle-to-grid communications," in *ITS Telecommunications (ITST), 2012 12th International Conference on*, Nov 2012, pp. 280–283.

[39] B. Liu and L. Zhang, "An Improved Identity-Based Batch Verification Scheme for VANETs," in *Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on*, Sept 2013, pp. 809–814.

[40] P. Vinoth Kumar and M. Maheshwari, "Prevention of Sybil attack and priority batch verification in VANETs," in *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*, Feb 2014, pp. 1–5.

[41] A. Perrig, "The BiBa One-time Signature and Broadcast Authentication Protocol," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, ser. CCS '01. New York, NY, USA: ACM, 2001, pp. 28–37.

[42] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," *Communications and Networks, Journal of*, vol. 11, no. 6, pp. 574–588, Dec 2009.

[43] "FIPS PUB 186-4: Digital Signature Standards."

[44] "OCRA: OATH Challenge-Response Algorithm," *RFC 6287*.

[45] 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.

[46] E. Mallada, X. Meng, M. Hack, L. Zhang, and A. Tang, "Skewless network clock synchronization," in *IEEE ICNP*, 2013.

[47] "Global Positioning System Standard Position Service (SPS) Performance Standard, 3rd edition," 2008.

[48] "RaspberryPi." [Online]. Available: www.raspberrypi.org

[49] H. Li, G. Dán, and K. Nahrstedt, "Portunes: Privacy-preserving fast authentication for dynamic electric vehicle charging," in *IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014*.

[50] H. Nicanfar, S. Hosseininezhad, P. TalebiFard, and V. Leung, "Robust privacy-preserving authentication scheme for communication between electric vehicle as power energy storage and power stations," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 3429–3434.

[51] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," in *IEEE SP*, 2003.

[52] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing Privacy Using Symmetric Random Key-Set in Vehicular Networks," in *ISADS*, 2007.

[53] Z. Gurkas Aydin, H. Chaouchi, and A. H. Zaim, "eHIP: early update for Host Identity Protocol," in *Mobility '09*.

[54] 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*.

[55] J. Li, H. Lu, and M. Guizani, "ACPN: A Novel Authentication Framework with Conditional Privacy-Preservation and Non-Repudiation for VANETs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 938–948, April 2015.

[56] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: An Efficient RSU-Aided Message Authentication Scheme in Vehicular Communication Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 1451–1457.

[57] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 9, pp. 1227–1239, Sept 2010.

[58] H. Nicanfar, P. TalebiFard, S. Hosseininezhad, V. C. Leung, and M. Damm, "Security and Privacy of Electric Vehicles in the Smart Grid Context: Problem and Solution," in *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, ser. DIVANet '13. New York, NY, USA: ACM, 2013, pp. 45–54.

[59] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," in *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET '07. New York, NY, USA: ACM, 2007, pp. 19–28.

[60] Z. Yang, S. Yu, W. Lou, and C. Liu, "P2: Privacy-Preserving Communication and Precise Reward Architecture for V2G Networks in Smart Grid," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 697–706, Dec 2011.

[61] M. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "Roaming electric vehicle charging and billing: An anonymous multi-user protocol," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 939–945.

[62] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," in *Advances in Cryptology CRYPTO 2004*, ser. Lecture Notes in Computer Science, M. Franklin, Ed. Springer Berlin Heidelberg, 2004, vol. 3152, pp. 41–55.

[63] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 6, pp. 3442–3456, Nov 2007.

[64] B. Chaurasia, S. Verma, and S. Bhasker, "Message broadcast in VANETs using group signature," in *Wireless Communication and Sensor Networks, 2008. WCSN 2008. Fourth International Conference on*, Dec 2008, pp. 131–136.

[65] X. Zhu, S. Jiang, L. Wang, H. Li, W. Zhang, and Z. Li, "Privacy-preserving authentication based on group signature for VANETs," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, Dec 2013, pp. 4609–4614.

[66] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient Privacy-Preserving Authentication for Vehicular Ad Hoc Networks," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 2, pp. 907–919, Feb 2014.

[67] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "An Efficient Message Authentication Scheme for Vehicular Communications," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3357–3368, Nov 2008.

[68] W. Shen, L. Liu, X. Cao, Y. Hao, and Y. Cheng, "Cooperative Message Authentication in Vehicular Cyber-Physical Systems," *Emerging Topics in Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 84–97, June 2013.

[69] X. Lin and X. Li, "Achieving Efficient Cooperative Message Authentication in Vehicular Ad Hoc Networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 7, pp. 3339–3348, Sept 2013.

[70] Y. Hao, T. Han, and Y. Cheng, "A cooperative message authentication protocol in VANETs," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 5562–5566.

[71] G. Yunchuan, Y. Lihua, L. Licai, and F. Binxing, "Utility-based cooperative decision in cooperative authentication," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1006–1014.

[72] "Petlib." [Online]. Available: https://github.com/gdanezis/petlib

[73] T. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," in *CRYPTO '91*.

[74] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proceedings on Advances in cryptology—CRYPTO '86*. London, UK, UK: Springer-Verlag, 1987, pp. 186–194.

[75] M. Dyer, R. Kannan, and J. Mount, "Sampling contingency tables," *Random Struct. Alg.*, vol. 10, pp. 487–506, 1997.

[76] A. Barvinok, "Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes," *Int. Math. Res. Notices*, pp. 348–385, 2009.

[77] R. G. Downey and M. R. Fellows, "Fixed-parameter tractability and completeness II: On completeness for W[1]," *Theor. Comput. Sci.*, vol. 141, no. 1-2, pp. 109–131, 1995.

[78] "VPriv: Protecting Privacy in Location-Based Vehicular Services," in *Presented as part of the 18th USENIX Security Symposium (USENIX Security 09)*. USENIX, 2009.

[79] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, "PrETP: Privacy-preserving Electronic Toll Pricing," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10.

[80] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham, "The Phantom Tollbooth: Privacy-preserving Electronic Toll Collection in the Presence of Driver Collusion," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11. USENIX Association, 2011.

[81] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," in *IEEE S&P '13*.

[82] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, 2014.

[83] J. Day, Y. Huang, E. Knapp, and I. Goldberg, "SPEcTRe: Spot-checked Private Ecash Tolling at Roadside," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, ser. WPES '11.

[84] D. Chaum, "Blind Signatures for Untraceable Payments," in *Advances in Cryptology*, D. Chaum, R. Rivest, and A. Sherman, Eds. Springer US, 1983, pp. 199–203.

[85] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, *Computer Security – ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Enhancing Location Privacy for Electric Vehicles (at the Right time), pp. 397–414.

[86] M. H. Au, W. Susilo, and Y. Mu, *Security and Cryptography for Networks: 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. Constant-Size Dynamic k-TAA, pp. 111–125.

[87] F. Kerschbaum, H. W. Lim, and I. Gudymenko, "Privacy-preserving billing for e-ticketing systems in public transportation," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, ser. WPES '13. ACM, 2013, pp. 143–154.