# Proactive key dissemination-based fast authentication for in-motion inductive EV charging

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

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

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

***Abstract*—In-motion inductive charging, or dynamic charging, is an emerging technology that allows electric vehicles (EVs) to be charged while on the move. Accurate billing for dynamic EV charging requires secure communication between the EVs and the utility, and could potentially require the secure delivery of small messages from the EVs to the utility at a very high rate, which is infeasible with the currently available solutions. In this paper we propose Fast Authentication for Dynamic EV Charging (FADEC) designed to meet the communication needs of in-motion inductive EV charging. FADEC features fast signing and verification, low communication overhead, and fast hand-off authentication to support EV mobility. Our simulations show that compared with ECDSA mandated by 802.11p standard, FADEC reduces data delivery delay by up to 97%, increases the data delivery ratio by more than an order of magnitude and enables timely data delivery even in a resource constrained environment.**

## I. Introduction

In-motion inductive charging, also known as dynamic charging [21], is a promising technology for charging electric vehicles (EV) while they move. The basic idea is to place charging coils under the charging pads on the road and attach charging coils to the EV's battery. When the EV is driving above the coil, the electromagnetic interaction between the coils under the road and the coils in the EV can charge the EV battery. Dynamic charging has received much attention from both the industry and the research community [3], [4], [8], [10], [21], and some companies have started testing the performance of dynamic charging on prototype buses [14].

The charge rate of dynamic charging depends on many factors, such as the distance between the coils, vehicle speed, and ultimately on the decision of the vehicle's driver whether to charge. Since the charge rate is not constant, dynamic charging can only become a commercial service if charged EVs can be billed accurately. Accurate billing requires that the EVs that should be billed can be identified, and that the EVs report their charging rate periodically to the utility providing the electricity. Fine grained billing under dynamic pricing and changing traffic conditions could potentially require the reporting to be very frequent. Real-time reporting from EVs also enables a variety of applications: 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.

Identification for the purpose of enforcement can be solved using smart cameras, as it is often done on toll roads, but reporting requires that there be Vehicle-to-Grid (V2G) communication between the EV and the utility. Since the communication between the EVs and the utility would serve for billing purposes, it is crucial that the utility authenticates the EVs' reports. Since reporting could potentially be very frequent, signing and verification of the reports should be fast.

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. DSRC is already used for electronic toll collection in many countries. 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. 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 IEEE 802.11p standard suggests the use of Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication in vehicular networks. Recent work [9] has shown, however, that using ECDSA it could take a significant 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 [9], [15], [16]. However, one-time signature is not the ideal solution in our scenario since it could incur non-trivial key generation and signing overhead [16], requires delayed verification [15], or puts restrictions on the content to be authenticated [9].

In this paper we propose *Fast Authentication for Dynamic EV Charging (FADEC)* designed to support the communication needs of dynamic wireless EV 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. To our best knowledge this is the first work that considers secure V2G communication for dynamic EV charging.

The rest of the paper is organized as follows. In Section II we introduce security background and review related work. In Section III we describe our system model and assumptions. In Section IV we describe the proposed authentication solution. We present simulation results in Section VI, and conclude our paper in Section VII.

## II. Background and Related Work

### A. Security Background

*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 [6].

*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 check 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 [9] 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.

*3) Just Fast Keying (JFK):* JFK [5] 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. 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.

### B. Related Work

Host Identity Protocol (HIP) [2] 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 [17] 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. While we have presented the initial idea of FADEC in a poster [12], the poster abstract contains very small part of the work presented in this paper.

Zhu et al. [22] 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.

Portunes [11] adopts a key pre-distribution approach combined with spatio-temporal verification for authentication between EVs and charging pads embedded in the roadbed within a contact duration of tens of milliseconds. In FADEC contact durations with the RSUs are longer, and FADEC disseminates the session key proactively as the EV moves along the road, which allows it to scale to larger areas.

Researchers have also considered one-time signatures for authentication in VANET. However, one-time signatures either require delayed verification [15], or incur non-trivial key generation overhead [16]. Time Valid HORS (TV-HORS) [20] combines one-way hash chains and HORS to reduce the frequency of public key distribution, and is robust against packet loss. The major drawback of TV-HORS is its public key size, which can be as large as 10KB. FastAuth [9] is proposed to authenticate vehicle safety messages of fixed format containing the location and the velocity of the vehicle, and generates short signatures by predicting the future locations of the vehicle. However, in our scenario the message content to be authenticated, i.e., the real-time statistics generated by the EV, might not be predictable, which makes FastAuth inapplicable.

## III. System Model

The system we consider 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.

**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.

**Attacker 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.

**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.

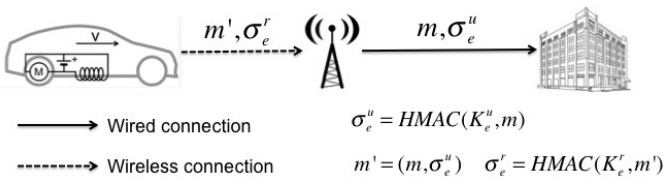


Fig. 1. Overview of FADEC.

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

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

*b) 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.

*c) 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.

## IV. 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. Fig. 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 $(m', \sigma_e^r)$ to the RSU. The RSU verifies the signature $\sigma_e^r$, 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 $\sigma_e^u$ 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 paper, although one could potentially use FADEC to establish another session key between the EV and the utility and use AES encryption.

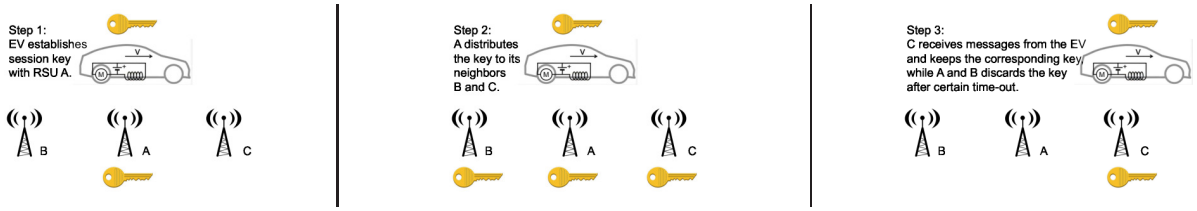


Fig. 2. Illustration of key establishment, dissemination to neighbors and discarding of unused keys.

### A. Establishing Session Key $K_e^r$ with the RSUs

The EV establishes its session key with the RSU using JFK [5]. 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 Fig. 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 Fig. 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 $\hat{t}_{A\to B} = \frac{d_{A\to B}^{max}}{v_{A\to B}^{min}}$, where $d_{A\to B}^{max}$ is the maximum travel distance to enter the range of $B$ from the range of $A$, and $v_{A\to B}^{min}$ 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 similarily.

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 $\lambda_A$ is the EV arrival rate at RSU $A$. The EV arrival rate $\lambda_A$ is bounded, and can be computed using results from traffic flow theory [13]. 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 $\rho_A$ (EVs/mile) then the arrival rate is $\lambda_A = \rho_A v_A$ [13]. 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 [22] 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 [22], 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.

### B. Establishing Session Key $K_e^u$ with the Utility

An EV establishes $K_e^u$ using JFK, but only after it has established $K_e^r$ 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.

### C. 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.

## V. Security Analysis

*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.

*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.

*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.

*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 obtainining $K_e^u$, 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.

*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.

*Privacy*: FADEC does not provide location privacy against the charging pad owner. A charging pad owner can easily follow EV movement through tracking license plates. It can, however, support encryption to hide the EV to utility communication from the charging pad owner.

## VI. 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 [7] 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 [19]. We use the mobility trace of 300 EVs as they traverse the 3kms 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 [18] 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 $\hat{t}_{A\to B} = 120$ s, and for each RSU $A$ we set $\hat{t}_A = 120$ s.

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 [1] 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 [6].

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.

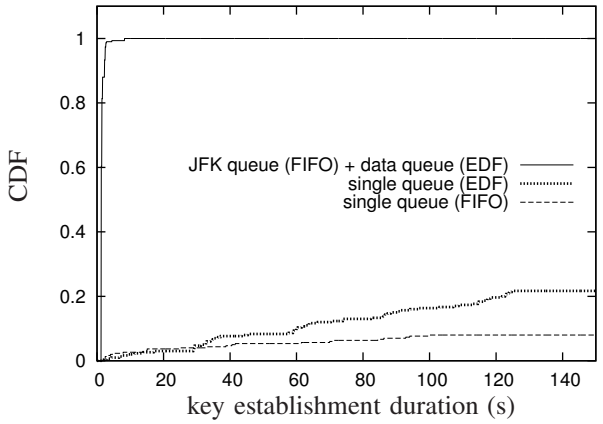


Fig. 3. Key establishment duration of the first 100 EVs in the resource constrained scenario with different RSU queue management strategies.

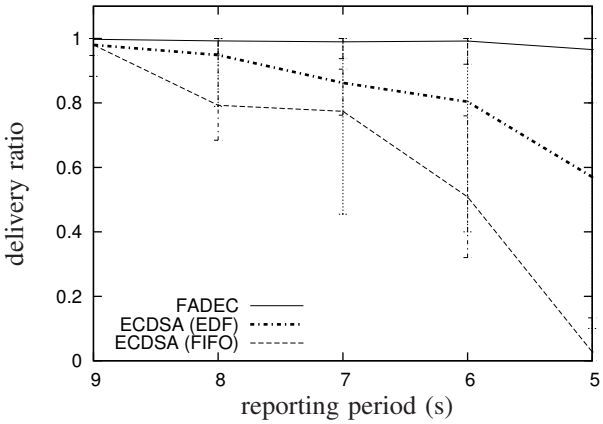


Fig. 4. Report delivery ratio under different reporting period in the resource constrained scenario.

### A. 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 neccesary 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 Fig. 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 succesful key establishment in FADEC when computational resources are scarce.

### B. Data Delivery Performance

*1) 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 Fig. 4 we show the delivery ratio as a function of the reporting period in the resource constrained scenario. We omit the results obtained in the resource rich scenario where both FADEC and ECDSA achieve delivery ratio close to 1. 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 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.

*2) 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 porportional to the delivery ratio.

In Fig. 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

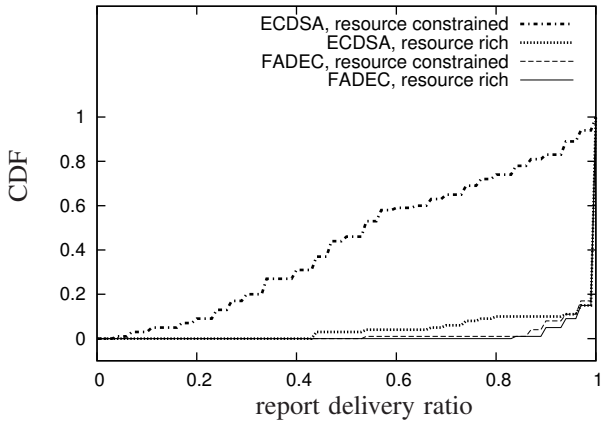


Fig. 5. Distribution of report delivery ratio.

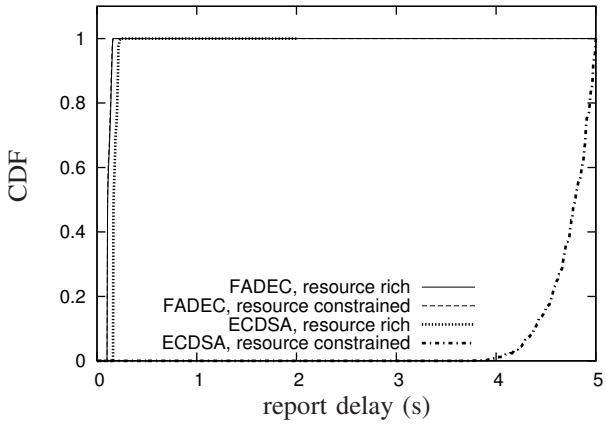


Fig. 6. Distribution of delay of successfully delivered reports.

makes the RSU data queue grow quickly, and most reports miss their deadlines even using EDF scheduling.

*3) Delay:* In Fig. 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 increases. This greatly affects the delay of ECDSA.

## VII. Conclusion

In this paper 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 hand-off 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.

## References


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

[2] Rfc 5201: Host identity protocol.

[3] Stanford report. http://news.stanford.edu/news/2012/february/wireless-vehicle-charge-020112.html.

[4] S. Ahn and J. Kim. Magnetic field design for high efficient and low emf wireless power transfer in on-line electric vehicle. In *Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference on*, pages 3979–3982, April 2011.

[5] 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.*, 7(2), May 2004.

[6] E. B. Barker and A. L. Roginsky. Sp 800-131a. transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. Technical report, Gaithersburg, MD, United States, 2011.

[7] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo - simulation of urban mobility: An overview. In *Proc. of SIMUL*, 2011.

[8] G. Covic and J. Boys. Modern trends in inductive power transfer for transportation applications. *Emerging and Selected Topics in Power Electronics, IEEE Journal of*, 1(1):28–41, March 2013.

[9] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer. Flooding-resilient broadcast authentication for vanets. In *Proc. of ACM MobiCom*, 2011.

[10] 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 *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, pages 1598–1601, Sept 2010.

[11] 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*.

[12] H. Li, G. Dán, and K. Nahrstedt. Fadec: Fast authentication for dynamic electric vehicle charging (poster abstract). In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 369–370, Oct 2013.

[13] I. Marsh. Vanet communication: A traffic flow approach. In *Proc. of IEEE PIMRC*, 2012.

[14] H. Perik. Practical EV Integration Cases for Static and Dynamic Wireless Power Transfer. In *International Energy Transfer for Electric Vehicles Conference*, 2013.

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

[16] L. Reyzin and N. Reyzin. Better than biba: Short one-time signatures with fast signing and verifying. In *ACISP*, 2002.

[17] J. So and J. Wang. Micro-hip a hip-based micro-mobility solution. In *ICC Workshops*, 2008.

[18] C. Sommer, R. German, and F. Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1), 2011.

[19] P. VARAIYA. What weve learned about highway congestion. *Access*, 27, 2005.

[20] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt. Time valid one-time signature for time-critical multicast data authentication. In *INFOCOM*, 2009.

[21] H. Wu, A. Gilchrist, K. Sealy, P. Israelsen, and J. Muhs. A review on inductive charging for electric vehicles. In *Electric Machines Drives Conference (IEMDC), 2011 IEEE International*, pages 143–147, May 2011.

[22] H. Zhu, R. Lu, X. Shen, and X. Lin. Security in service-oriented vehicular networks. *IEEE Wireless Communications*, 16(4):16–22, 2009.