FADEC: Fast Authentication for Dynamic Electric Vehicle Charging

Hongyang Li
UIUC
hli52@illinois.edu

György Dán
KTH
gyuri@kth.se

Klara Nahrstedt
UIUC
klara@illinois.edu

***Abstract*—Dynamic wireless 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 power utility, and could potentially require the secure delivery of small messages from the EVs to the utility at a very high rate, which is an infeasible task with the currently available solutions. In this paper we propose Fast Authentication for Dynamic EV Charging (FADEC) designed to meet the communication needs of dynamic EV charging. FADEC features fast signing and verification, incurs low communication overhead, and fast hand-off authentication to support EV mobility. Our simulations show that compared with ECDSA mandated by the 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

Dynamic wireless charging [1], [2] is a promising technology for charging electric vehicles (EV) while driving. 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. The charge rate 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 charged can be identified, and that the EVs report their battery levels periodically to the utility company providing the electricity. Fine grained billing under dynamic pricing and changing traffic conditions could potentially require the reporting to be very frequent.

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 communication between the EV and the utility company. Since the communication between the EVs and the utility would serve for billing purposes, it is crucial that the utility can authenticate the EVs' reports. Since reporting could potentially be very frequent, the signing and the verification of the EV 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 company, 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 company 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 [3] 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 overcome the disadvantage of computation overhead of ECDSA, researchers have proposed the use of one-time signature for authentication [3]–[6]. However, one-time signature is not the ideal solution in our scenario since it could incur non-trivial key generation and signing overhead [4], requires delayed verification [5], or put restrictions on the content to be authenticated [3].

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 have to re-authenticate itself every time it encounters a new RSU. Our simulations show that FADEC is suitable for dynamic EV charging scenarios. Compared with ECDSA, FADEC reduces the data delivery delay by at most 97% and improves the delivery ratio by more than an order of magnitude.

The rest of the paper is organized as follows. In Section II we introduce security background and reveiw related work, and 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 V, and conclude our paper in Section VI.

II. Background and Related Work

A. Security Background

1) HMAC: Hash-based Message Authentication Code (HMAC) is a fast authentication mechanism based on symmetric keys. The sender and receiver both have the same symmetric key k, and 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 $HMAC(k, M') = HMAC(k, M)$. The advantage of HMAC authentication is its fast signing and verification speed.

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 else 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 an improved version of DSA. The IEEE 802.11p standard suggests the use of ECDSA to authenticate vehicle safety messages. However, previous work [3] 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): Just Fast Keying (JFK) [7] 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. 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

TESLA [5] is a broadcast authentication protocol that features fast message signing and signature verification. It uses one-way hash function to form a chain of symmetric keys and sign each message using one key from the key chain. The message with its signature is sent first, but the key used to sign the message is revealed later to the receiver. This delayed verification is the major drawback of TESLA, and makes it infeasible for our real-time statistics reporting scenario.

HORS [4] is a representative one-time signature scheme with fast signing and verifying. However, the major drawback of HORS is that, every time the sender wants to sign a message, it needs to generate lots of public/private key pairs, and distribute the public keys to the receivers. In our dynamic wireless charging scenario, the EV is generating messages with real-time statistics and responsible for signing these messages, and the large key generation overhead makes it impractical to apply HORS in our scenario. Time Valid HORS (TV-HORS) [6] 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 [3] is proposed to authenticate vehicle safety messages that include 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 and Design Considerations

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, a utility company that provides power to the pad, and the EVs.

We assume that each EV is equipped with a DSRC on-board unit and can communicate wirelessly with the RSUs. The RSUs and the utility company are connected through a backbone network. In order to communicate with the utility company, the EV will send its messages wirelessly to an RSU, which will then relay the EV's messages to the utility company. If the utility company 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 a full deployment of PKI, i.e., the EVs, the RSUs, and the utility company 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 company and each RSU. FADEC thus focuses mainly on the authentication between the EVs and the RSUs, and between the EVs and the utility company. We assume that all EVs and all RSUs have similar limited computational resources to sign messages and to verify signatures, while the utility company has significantly more computational resources. We assume that the attacker has similar computational resource as an ordinary EV or RSU. We assume the attacker could compromise an arbitrary number of EVs and obtain all their secrets including the private keys and the established session keys, but cannot compromise the CA, the utility company, or any RSU.

An EV could potentially turn off its on-board unit in an attempt to charge the battery without being billed. One way to prevent this is to place smart 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 can be levied a fine. This provides an incentive for an EV to initiate communication with the RSUs and with the utility company.

The ultimate goal for FADEC is to allow the utility company to verify the integrity of messages sent by an EV and the identity of its sender for correct billing. This is not enough, however. Without further authentication, an attacker could impersonate an RSU or the utility company to attract 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 charging scenario also requires that the EV can authenticate the identity of the utility company before sending real-time reports, to ensure the reports are delivered to the proper utility. Since all messages between the EV and the utility company 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.

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

a) Fast message signing: The EVs might need to send charging-related information to the utility company at a high rate. Therefore they should be able to sign messages quickly.

b) Fast signature verification: Since the RSU and the utility company may handle messages from many EVs and must verify many signatures, verification has to be fast. Fast signature verification is also important to defend against DoS attacks. If the verification takes long, an attacker could launch a DoS attack by flooding the network with fake signatures, and the recipient RSU or the utility company would spend most or all of its computational resources on verifying the fake signatures.

c) Fast hand-off authentication: As the EV moves, the RSU associated with the EV can change. 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.

d) Low communication overhead: The signature length must be short. This requirement is motivated by that an EV will most likely generate many messages of small sizes, e.g., messages containing only its battery state of charge. Attaching a long signature to a short message means more energy is used to transmit the signature instead of the message content.

IV. FADEC

In the following we describe the proposed FADEC authentication scheme. We illustrate FADEC in Fig. 1. EV e establishes a symmetric session key K_e^r with the RSUs and another symmetric session key K_e^u with the utility company. The session keys are established using JFK. Before sending a

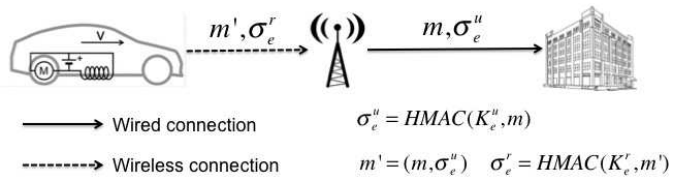


Fig. 1. Overview of FADEC.

message m [1] to the utility company, the EV 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 company through the previously established secure channel. The utility company verifies the signature σ_e^u and then accepts the message m. In the following section we describe how the EV establishes the two session keys K_e^r and K_e^u.

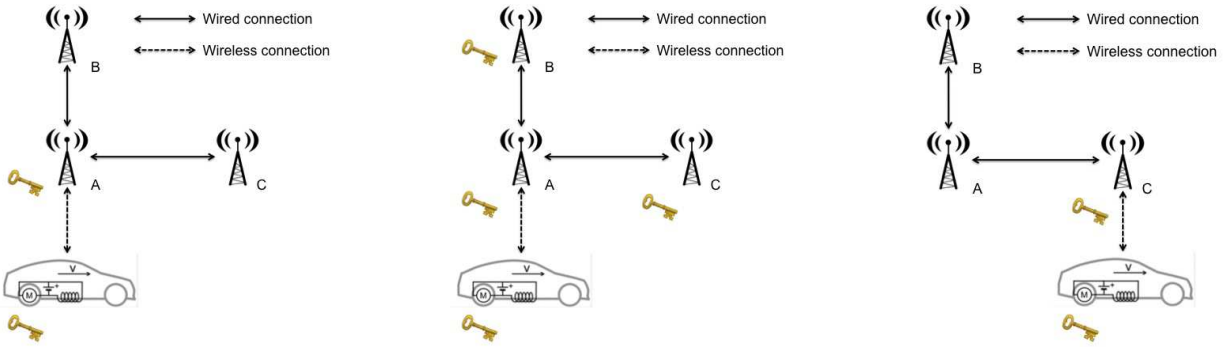
A. Establishing Session Key K_e^r with the RSUs

The EV establishes a session key with RSU using JFK [7]. The challenge here is that, 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. One naive approach is 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.

FADEC follows the approach of key dissemination. The EV establishes a session key K_e^r with its first encountered RSU using the JFK protocol, and will authenticate with all subsequent RSUs using the same key K_e^r. The challenge is to ensure that by the time the EV enters the communication range of a new RSU, that RSU has already obtained the corresponding session key K_e^r from the RSU previously associated with the EV. Flooding the key is infeasible as it incurs too much communication overhead. An alternative is for the current RSU to predict the movement of the EV and send the key to the RSU that the EV will most likely encounter. This approach, however, unnecessarily complicates the design of the authentication framework, and the EV mobility could be difficult to predict given complex road topology.

We solve the key dissemination problem using a broadcast-and-discard approach, 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 $t_{A\to B}$ seconds, where $t_{A\to B}$ is a fixed parameter that estimates the maximum time required for an EV currently in range of RSU A to move into the range

[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 company and use AES encryption.



Step 1: EV e establishes key K_e^r with the current RSU A.

Step 2: A disseminates K_e^r to both neighbor RSUs B and C.

Step 3: When EV e enters the range of C, C keeps the key K_e^r, while both A and B discard the key.

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

of B.[2] If EV e does not try to communicate with RSU B using K_e^r within $t_{A\to B}$ time then RSU B discards the key. Similarly, when C receives K_e^r, it stores the key for $t_{A\to C}$ seconds. In Fig. 2, EV e is moving towards C, and enters the range of C within $t_{A\to C}$ seconds. If EV e communicates with RSU C using K_e^r, then C will store K_e^r for additional t_C seconds, where t_C is a fixed parameter estimating the duration when EV e stays within the range of C. [3] Intuitively, the number of keys stored by an RSU is very small, since an EV generally stays within its range for tens of seconds. Our simulations confirm this; in a heavily loaded highway scenario each RSU needs to hold about 30 keys on average.

Note that an RSU will disseminate the key K_e^r to its neighbor RSUs only when the corresponding EV starts communicating with it. In Fig. 2, if the EV moves into the range of RSU C but does not send any message, then C will also discard the key and will not disseminate the key any further. When the EV moves to the next RSU beyond C, that RSU might not have the key K_e^r. However, in our scenario where EVs send real-time reports periodically, it is unlikely that an EV will not send any message to the RSU while it stays within its range. Furthermore, even if the current RSU does not have the key, it can establish a new session key with the EV.

Since packet loss is common in vehicular networks [8], the EV may not be able to finish a key establishment process with the RSU or with the utility company in one attempt. Therefore, we employ an exponential backoff strategy. In our simulations we found that exponential backoff with an initial timeout of 1 second and backoff multiplier of 2 achieved good performance.

B. Establishing Session Key K_e^u with Utility Company

An EV starts establishing K_e^u only when it has established K_e^r with the RSU. K_e^u is also established using JFK. Since the EV cannot directly communicate with the utility company, it has to send its messages to an RSU, and the RSU will relay the messages to the utility company. Once the EV has established K_e^r with the RSUs, it will sign its messages using K_e^r before sending them to the RSU, and the RSU will verify the signature before relaying the EV's messages. When the utility company 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., p2p file sharing) at the same time. The existence of such background data 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 RSU could delay the processing of JFK messages indefinitely.

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, a key establishment message will never be delayed because some application messages have arrived before it. 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. Performance Evaluation

We simulate road traffic on a 4-lane single-direction straight road segment of 3km. There are 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 [9] 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 [10]. 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

[2] $t_{A\to B}$ can be effectively estimated given the locations of A, B and the road map. For example, let l_A and l_B be the locations within the range of RSU A and B respectively, such that the length d of the road connecting l_A and l_B is maximized. If the road has a lower speed limit of v m/s, a reasonable estimation could be $t_{A\to B} = d/v$.

[3] t_C can be effectively estimated given the communication range of C and the road map. In the case of a single straight road, a reasonable estimation is $t_C = 2R/v$, where R is the communication range and v is the lower speed limit.

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 company and each RSU, and between each pair of neighbor RSUs. The propagation delay between the utility company and each RSU is set to 100 ms, and the delay between neighbor RSUs is set to 1 ms. We use the Veins [11] 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.

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.

In all our simulations the EVs generate 1024 bits of information per second. Unless otherwise noted, each EV sends a report to the utility company 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.

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 company. The successful establishment of K_e^u thus implies the establishment of K_e^r. In Fig. 3 we show the distribution of the time taken to establish K_e^u when the RSUs prioritize key establishment message processing as decribed in Section IV-C. The figure shows that 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 company). The one-time cost of 8.3 second is small compared to the time scale in a dynamic EV charging scenario. [4]

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. 4 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

[4]In our simulation, it takes around 144 seconds for an EV to travel through the 3 km road segment, during which time it spends, only once, a maximum of 8.3 seconds to establish session keys. Even at high speed of 105 km/h (65 mph), an EV still needs about 103 seconds to travel through the road segment.

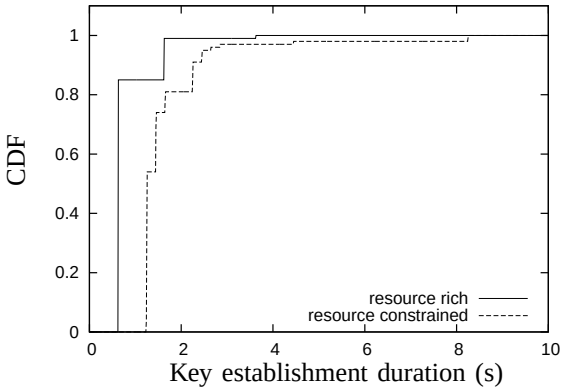


Fig. 3. Distribution of establishment duration for EV-utility key K_e^u across all EVs.

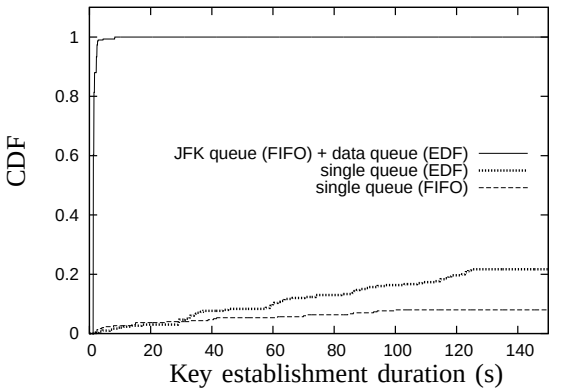


Fig. 4. Establishment duration for EV-utility key K_e^u of the first 100 EVs in the resource constrained scenario with different RSU queue management strategies.

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 JFK queue greatly reduces the key establishment duration, and allows all EVs to complete the key establishment with both the RSU and the utility company within 8.3 seconds. 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 wireless 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 company 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 considered as delivered successfully if it arrives at the utility company within t seconds. Each report contains all information generated by the EV since the last report sent. A large reporting period t means the EVs send reports less often, but each report is larger

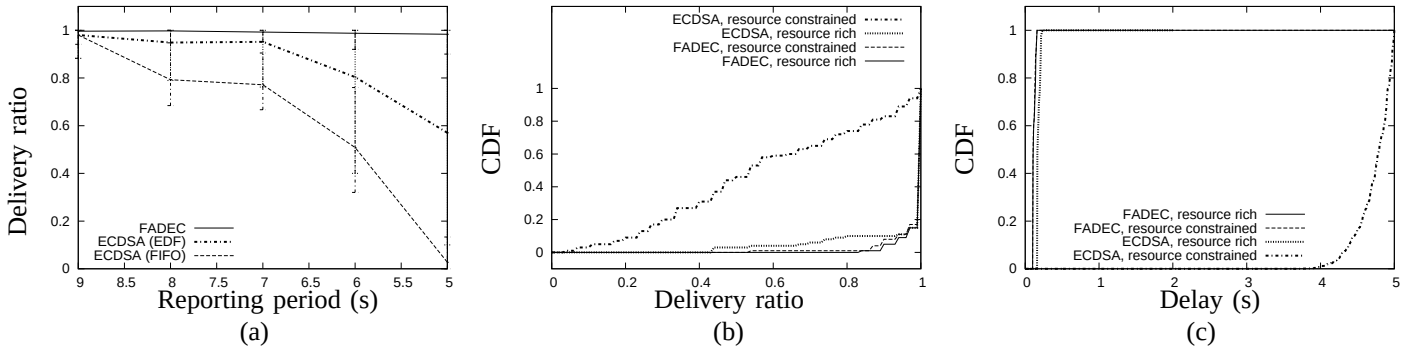


Fig. 5. Data delivery performance of FADEC, report delivery ratio and delay under various scenarios.

as it contains more information.

In Fig. 5 (a) 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. For the rest evaluations 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 company to obtain up-to-date information about the EV status. In our scenario where all EVs send their reports at the same frequency, throughput is porportional to the delivery ratio.

In Fig. 5 (b) 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% 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 will miss their deadlines even using EDF scheduling.

3) Delay: Finally, we consider the data delivery delay observed by the utility company. 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 company to verify the signature. This is an important metric for our evaluation, since a shorter delay means the utility company 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.

In Fig. 5 (c) we plot the distribution of the delay of all reports that successfully arrived at the utility company within their deadlines. 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.

VI. 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 company, and achieves fast signing, fast verification, fast hand-off authentication, and low communication overhead. Our simulations have shown that FADEC 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] "Stanford report," http://news.stanford.edu/news/2012/february/wireless-vehicle-charge-020112.html.

[2] X. Yu, S. Sandhu, S. Beiker, R. Sassoon, and S. Fan, "Wireless energy transfer with the presence of metallic planes," *Applied Physics Letters*, vol. 99, no. 21, 2011.

[3] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Flooding-resilient broadcast authentication for vanets," in *MobiCom*, 2011.

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

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

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

[7] 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.

[8] 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.

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

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

[11] 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.