Lynx: Authenticated Anonymous Real-Time Reporting of Electric Vehicle Information

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*—With the proliferation of electric vehicles (EVs), EV battery charging will be a significant load on the power grid, and thus it will have to be optimized. Many proposed optimizations rely on predicted EV information, such as future trip start time and battery State-of-Charge (SoC), for load management, charging scheduling, V2G profit optimization, etc. Prediction in turn would benefit from real-time information about the EVs, while they are on the road, i.e., before arriving at the charging facility. A real-time reporting framework is thus necessary so that EVs can report status information to the utility in a timely fashion. In this paper we present *Lynx*, an authenticated anonymous real-time reporting protocol. Lynx allows an EV to send anonymous reports using pseudonyms that are unlinkable to its true identity. At the same time, the utility can verify that the reports come from *some* authentic EV without knowing the exact identity of the EV that sends the report. To encourage EV participation, Lynx allows the utility to issue anonymous receipts to EVs, which can be used later by the EVs to anonymously claim credits. Lynx minimizes computation overhead during real-time reporting, and our implementation on Raspberry Pi 2 shows that report generation and verification can be done within 10 ms.**

I. Introduction

Charging electric vehicles (EVs) will put a significant load on the future power grid. To ensure efficient grid operation, there is an emerging consensus that EV charging/discharging will have to be coordinated, in order to reduce power losses caused by uncontrolled EV charging [1], to avoid congestion in the distribution network [2], [3], to adjust real-time electricity price [4], to minimize the EVs' waiting times at charging stations [5], and to optimize V2G energy and ancillary service scheduling [6], etc. A prerequisite for efficient coordination is accurate prediction of the EVs' activities, such as arrival, departure, and battery State-of-Charge (SoC), which requires the EVs to report real-time status information to the utility upon request.

To gain acceptance, the real-time reporting of EV information must be secure, efficient, and privacy-preserving. The utility must be able to authenticate the reports to make sure that it only uses reports from authorized EVs. At the same time, since the report may contain sensitive information (e.g., the EV's current location), the EV has a natural incentive to remain anonymous. Real-time reporting should also have a low communication overhead to be bandwidth efficient. Last but not least, the utility may want to reward the EVs for the information provided, and thus the reporting system should support the implementation of an incentive mechanism.

The most straightforward approach would be to let each EV submit its own report directly to the utility using digital signature authentication. However, digital signature is computationally expensive [7] and identity-revealing. Several works [8], [9] improve authentication efficiency by using symmetric keys, but do not provide anonymity protection. Portunes [10] uses utility-issued pseudonyms to protect the EV's privacy from third-party entities, but the utility knows the mapping between pseudonym and EV's true identity. Token-based approaches [11], [12] let the EV authenticate itself by revealing an authentication token to the utility, but the token revealing process is computationally intensive and is not efficient for real-time reporting (e.g., in dynamic charging scenario [10] the EV needs to send a report to a new charging pad every tens of milliseconds).

In this paper, we propose *Lynx*, an efficient and secure anonymous real-time reporting protocol for EVs to submit real-time reports to the utility. Lynx achieves efficient real-time reporting by performing computationally intensive pseudonym and session key allocation during the night when the EV is parked at home, which allows the EV to use efficient symmetric key authentication for real-time reporting on the road without invoking additional key exchange protocols. To protect the EV's anonymity, Lynx uses partially blind signatures and an anonymous authenticated version of the Diffie-Hellman key exchange protocol during the session key negotiation so that the utility does not know the mapping between the EV's true identity and its pseudonym/session key. Finally, to support various incentive mechanisms, Lynx allows the utility to issue anonymous receipts to EVs, and these receipts do not reveal any information about the EV's identity or about the task that the EV has performed (otherwise the utility could advertise a special task only to a specific area, and a receipt showing that the EV has performed that particular task gives away the information that the EV was at that area).

The rest of our paper is organized as follows: in Section II we review security background and related work; in Section III we describe the system model and design goals of Lynx; in Section IV we present the Lynx protocol; in Section V we analyze security and privacy properties of Lynx; in Section VI we evaluate the performance of Lynx, and conclude our paper in Section VII.

II. Security Background and Related Work

A. Security Background

Authenticated Encryption: Encryption and Message Authentication Code (MAC) can be combined in different ways to provide both message integrity and authenticity. In Encrypt-then-MAC (EtM) the sender sends $E[m], \text{MAC}(E[m])$, where

978-1-4673-8289-2/15/$31.00 ©2015 IEEE

the plaintext is first encrypted, and the MAC on the ciphertext is appended. EtM was proved to be more secure than both MAC-then-Encrypt (MtE) and Encrypt-and-MAC (E&M) [13], and thus Lynx adopts EtM for real-time reporting.

Blind Signature: Blind signature [14] allows the user to request a signature of a message from the signer without the signer learning about the content of the message. The signature requester generates a secret pair of blinding/unblinding operations (b, b^{-1}), and applies the blinding operation b to the plaintext message m. The requester then sends the blinded message $b(m)$ to the signer. The signer signs the blinded message with operation s and produces a signature $s(b(m))$, and returns the signature to the requester. The requester now applies the unblinding operation b^{-1} to the signature to obtain $b^{-1}(s(b(m))) = s(m)$, which is a signature on the plaintext message. Note that the signer only knows the blinded message $b(m)$, not the plaintext message m. Only the original requester can unblind a signature $s(b(m))$ to obtain $s(m)$. The requester can further verify that $s(m)$ is indeed a valid signature on m, but cannot forge such a signature.

Partially Blind Signature: Partially blind signature (PBS) [14] is similar to blind signature in that the signer does not learn the content of the signed message. However, PBS allows the signer to include some common message in the signature. The signer and the requester first agree on the content of the common message m_0. The requester submits blinded message $b(m)$, the signer generates signature $s(b(m), m_0)$, and returns it to the requester. The requester applies the unblinding operation to obtain $b^{-1}(s(b(m), m_0)) = s(m, m_0)$.

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 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 [15] or incorporating the public key of the intended communicating parties in the shared secret [16]. As a result IA-DHKE does not provide anonymity. Lynx adopts the idea of IA-DHKE but instead allows the EV to use an authorization token τ to anonymously authenticate itself to the utility.

B. Related Work

AnonySense [17] is a general platform for anonymous task allocation and report submission. To achieve anonymous reporting, the participant submits its reports through a MIX network. In order to defeat timing attacks on MIX network, AnonySense abandons the use of low-latency anonymizing networks (e.g., Tor [18]), and uses instead Mixmaster [19], which delays sending messages until it collects enough messages and can mix them reliably. The delayed reporting thus makes it difficult to apply AnonySense in our scenario where timely information is a key requirement. PEPSI [20] assumes that the participating mobile nodes and the querier do not share a common secret, and thus chooses an Identity-Based Encryption (IBE) approach for anonymous reporting. In our scenario it is reasonable to assume that the utility and the EV know each other (i.e., they know the public key of each other). With this assumption Lynx is able to establish anonymously a session key and uses efficient symmetric cryptography to authenticate real-time reports. SPPEAR [21] uses oblivious transfer [22] to allow an EV e to obtain an authorization token τ from a set of tokens Π_e generated by the utility, and then uses the authorization token to establish a pseudonym. The use of oblivious transfer unnecessarily complicates the protocol design, as the protocol must guarantee that (i) for any two different EVs e and e', the corresponding token sets $\Pi_e, \Pi_{e'}$ from which they obtain the token have non-empty intersection, i.e., $\Pi_e \cap \Pi_{e'} \neq \emptyset$; and (ii) no two EVs can obtain the same token. Anonymous e-Tokens [12] allows a participant to show the token at most n times anonymously, where n is a system parameter. If the participant attempts to show the token more than n times, its identity can be inferred. However, the verification of e-Token requires an online zero-knowledge proof, which does not satisfy our efficiency requirement due to its computation and communication overhead. Li and Cao [11] proposed a privacy-aware incentive mechanism using token-based authentication for real-time reporting, but their design requires the EV to reveal all unused tokens at the end of each iteration, which incurs additional communication overhead.

Lynx differs from previous works in that it allows the computationally intensive part of the protocol to be performed a priori. This design choice is motivated by the common observation that most urban cars are parked at night (e.g., 1 am - 5 am) [23], which allows the EV to perform expensive cryptographic and network operations such as communicating through Mixmaster [19] or revealing anonymous tokens. Lynx is also novel in that it relies on an anonymous token for IA-DHKE in establishing pseudonyms and session keys, which are thus unlinkable to the EV's true identity.

III. Models and Goals

A. System Model

We consider a scenario where the utility needs the information about the EV, such as its current SoC, desired SoC, and estimated time-of-arrival, before the EV arrives at the charging facility. Thus, reporting should be possible while the EV is on the road, and we assume that the EVs are able to communicate wirelessly with the utility either through a wireless network (WiFi/DSRC or cellular) while on the road.

We assume that the EV follow the typical time-of-day pattern of urban cars, and will be parked at night for several hours [23]. While parked at night, the EV can perform the computationally intensive part of the Lynx protocol to establish pseudonyms and session keys, so that during the day it can readily use efficient symmetric encryption and MAC authentication for real-time reporting.

B. Security Model

We assume each EV e has a public/private key pair (P_e, S_e), and the utility has public/private key pair (P_U, S_U). The utility and EV know the public key and the corresponding public key certificate of each other. We also assume that the EVs and the utility have agreed on a base g for Diffie-Hellman key exchange.

We assume that a secure partically blind signature (PBS) scheme (e.g., [24]) is available. We assume that the utility has two different key pairs (P_τ, S_τ), (P_γ, S_γ) for PBS generation/verification: (P_τ, S_τ) is used only for authorization token τ, and (P_γ, S_γ) is used only for receipt γ. S_τ and S_γ are kept secret by the utility, while P_τ and P_γ are known to all EVs. In both cases we assume that the common message m_0 appended to the PBS is the current date in the format mm/dd/yyyy. We further assume that many EVs request authorization token τ and receipt γ each day, in which case the appended common message m_0 will not compromise the EV's anonymity.

We assume the attacker is computationally bounded. In particular the attacker cannot reverse a one-way hash or forge digital signatures without the private key. The attacker may compromise one or multiple EVs, and access all secrets of compromised EVs including their secret keys for digital signature. The attacker may also compromise the utility and access all its secrets including the private key S_U for digital signatures and the signing keys S_τ and S_γ for blind signatures.

C. Goals

Efficiency: Lynx should be bandwidth-efficient and computationally lightweight on the EVs, due to their limited computation resources.

Anonymity: Lynx should protect the EVs' anonymity during real-time reporting. Consider the example where the utility wants to learn about the locations of all the EVs with less than 30% battery. If the EV uses its true identity (e.g., long-term public key) to authenticate and send the reports, the utility will be able to know the EV's location at a particular time, which compromises the EV's privacy.

Authorization: Lynx should ensure that only authorized EVs can submit reports, but without revealing the EVs' identities. Authorization avoids an attacker from submitting fake reports that could influence the utility's decision.

Reward-compatibility: The utility may want to reward the EVs for submitting reports. To facilitate rewarding schemes, Lynx should allow the utility to issue unforgeable receipts whose validity the utility should be able to verify. At the same time, the receipt should not be linkable to the report for which it was issued for.

IV. Design

Lynx consists of four phases: token acquisition, key establishment, real-time reporting, and receipt submission. In the token acquisition phase the EV acquires multiple authorization tokens $\tau_i, i = 1 \ldots N$, where N is the maximum number of pseudonyms that an EV can acquire each day. During the key establishment phase, the EV uses each authorization token as an anonymous credential to establish a pseudonym π_i and two session keys k_i^E and k_i^A. The pseudonyms (session keys) are unlinkable to other pseudonyms (session keys), in that the utility cannot tell whether two pseudonyms (session keys) belong to the same EV. In the real-time reporting phase, the EV chooses randomly an unused pseudonym π_i to submit a report and uses the corresponding session keys k_i^E, k_i^A for encryption and authentication. Each pseudonym is used only once to protect the EV's location privacy. Finally, the EV submits receipts to the utility in order to get rewarded.

σ_A	A's digital signature the entire message
N	total number of pseudonyms issued to EV e. for all subscript i in this table we have $1 \leq i \leq N$.
$K_i[m]$	symmetric encryption using key K_i on m
$C(x, z)$	Pedersen commitment of x with opening secret z
π_i	pseudonyms issued by the utility to an EV
$\text{EtM}_{k_i^E}^{k_i^A}\{m\}$	Encrypt-then-MAC: uses AES to encrypt m with key k_i^E, then computes the MAC on the encrypted message with key k_i^A
$\text{PBS}_S^{m_0}(m)$	partially blind signature on m with key S and appended common message m_0
m_0	current date (mm/dd/yyyy)
τ_i	authorization token $\tau_i = \text{PBS}_{S_\tau}^{m_0}(C(g^{x_i}, z_i))$
γ_i	EV's receipt $\gamma_i = \text{PBS}_{S_\gamma}^{m_0}(C(acc_i, r_i))$
acc_i	account information specifying how the utility should pay the EV with pseudonym π_i.
(P_τ, S_τ)	public/secret key pair to generate and verify all tokens τ_i. S_τ is known only to the utility.
(P_γ, S_γ)	public/secret key pair to generate and verify all receipts γ_i. S_γ is known only to the utility.
b_e, b_e^{-1}	EV e's blinding/unblinding operation

TABLE I
NOTATIONS

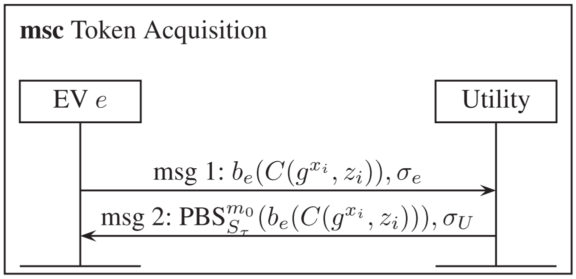
msc Token Acquisition
EV e
Utility
msg 1: $b_e(C(g^{x_i}, z_i)), \sigma_e$
msg 2: $\text{PBS}_{S_\tau}^{m_0}(b_e(C(g^{x_i}, z_i))), \sigma_U$

Fig. 1. Token acquisition in Lynx. The EV repeats this phase N times to obtain tokens $\tau_i, i = 1 \ldots N$.

Token acquisition, key establishment, and receipt submission are computationally intensive, but they do not need to be performed in real-time. For instance, the EV may acquire tokens and establish session keys when parked at home during the night so that it can readily use the pseudonyms and session keys for real-time reporting during the next day.

All messages include a timestamp to defend against replay attacks. To simplify presentation we omit the timestamp in the message specifications. In Table I we summarize the notations.

A. Token Acquisition

Each EV e runs the token acquisition protocol N times to establish N tokens $\tau_i, i = 1 \ldots N$. For each i, the EV generates two random secrets x_i and z_i, computes g^{x_i}, and commits g^{x_i} with opening secret z_i in $C(g^{x_i}, z_i)$ using a secure commitment scheme (e.g., [25]). It then blinds the commitment using b_e,

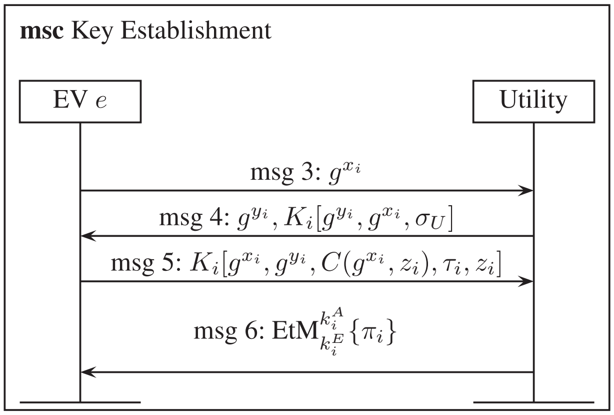
msc Key Establishment
EV e
Utility
msg 3: g^{x_i}
msg 4: $g^{y_i}, K_i[g^{y_i}, g^{x_i}, \sigma_U]$
msg 5: $K_i[g^{x_i}, g^{y_i}, C(g^{x_i}, z_i), \tau_i, z_i]$
msg 6: $\text{EtM}_{k_i^E}^{k_i^A}\{\pi_i\}$

Fig. 2. Key establishment in Lynx. The EV repeats this phase N times to obtain $(\pi_i, k_i^E, k_i^A), i = 1 \ldots N$.

signs the message with its true identity (e.g., using ECDSA), and sends

$$\text{msg 1: } b_e(C(g^{x_i}, z_i)), \sigma_e \tag{1}$$

to the utility, where $b_e(C(g^{x_i}, z_i))$ is the blinded commitment and σ_e is EV e's digital signature on the entire message.

The utility verifies the digital signature σ_e on the message. Note that σ_e reveals EV e's true identity. The utility generates a partially blind signature $\text{PBS}_{S_\tau}^{m_0}(b_e(C(g^{x_i}, z_i)))$ where m_0 is the appended common message, which is the current date in the format mm/dd/yyyy. It is reasonable to assume that many EVs will request tokens each day, and thus m_0 does not reveal the identity of the EV. The utility then returns

$$\text{msg 2: } \text{PBS}_{S_\tau}^{m_0}(b_e(C(g^{x_i}, z_i))), \sigma_U \tag{2}$$

to the EV, where σ_U is the utility's digital signature on the entire message.

When EV e receives msg 2, it verifies σ_U, and then applies the unblinding operation b_e^{-1} to obtain the token

$$\tau_i = b_e^{-1}(\text{PBS}_{S_\tau}^{m_0}(b_e(C(g^{x_i}, z_i)))) = \text{PBS}_{S_\tau}^{m_0}(C(g^{x_i}, z_i)) \tag{3}$$

and verifies that τ_i is a valid signature on $C(g^{x_i}, z_i)$ using public key P_τ. Note that the utility does not know the signed content $C(g^{x_i}, z_i)$ at this point. In Fig. 1 we illustrate the token acquisition phase.

B. Key Establishment

For each token τ_i acquired, the EV runs the key establishment protocol shown in Fig. 2 to obtain a pseudonym π_i and to establish two session keys k_i^E, k_i^A with the utility. However, the key establishment phase does not have to *immediately* follow the corresponding token acquisition phase. In fact, to defeat timing-based inference, after the EV acquires the token, it should wait for a random period of time before starting key establishment. To hide the EV's network identity, all messages in this phase should be sent through an anonymizer (e.g., Tor [18] or Mixmaster [19]). Note that since key establishment is generally performed when the EV is parked at night and has several hours to complete the phase, the EV can afford to use a high latency anonymizer such as Mixmaster [19] that provides resistance against timing attacks.

To initiate key establishment the EV sends to the utility

$$\text{msg 3: } g^{x_i} \tag{4}$$

Note that the EV has committed the value of g^{x_i} in τ_i.

When receiving msg 3, the utility generates a random secret y_i, constructs the shared session key

$$K_i = (g^{x_i})^{y_i} = g^{x_i y_i} \tag{5}$$

and sends to the EV

$$\text{msg 4: } g^{y_i}, K_i[g^{y_i}, g^{x_i}, \sigma_U] \tag{6}$$

where σ_U is the utility's digital signature on the ordered pair (g^{y_i}, g^{x_i}).

After receiving the utility's response, the EV constructs the shared session key $K_i = (g^{y_i})^{x_i} = g^{x_i y_i}$, decrypts the rest of msg 4 using K_i, and verifies the utility's digital signature σ_U on (g^{y_i}, g^{x_i}). At this point the EV has authenticated the utility.

The EV proceeds to show the token τ_i it obtained from msg 2 by sending the following message to the utility:

$$\text{msg 5: } K_i[g^{x_i}, g^{y_i}, C(g^{x_i}, z_i), \tau_i, z_i] \tag{7}$$

Now the utility uses K_i to decrypt the message, obtains the secret z_i, and verifies (i) g^{x_i}, g^{y_i} are valid; (ii) z_i opens the commitment $C(g^{x_i}, z_i)$; (iii) τ_i is a valid partially blind signature on $C(g^{x_i}, z_i)$ with appended common message m_0 and signed using key S_τ; and (iv) the date specified by m_0 is the same as the current date.

From the shared key K_i, both EV e and the utility derive two shared keys k_i^E, k_i^A, where k_i^E is used for symmetric encryption and k_i^A is used for MAC computation. The utility generates pseudonym π_i, and sends to the EV

$$\text{msg 6: } \text{EtM}_{k_i^E}^{k_i^A}\{\pi_i\} \tag{8}$$

C. Real-Time Reporting

In the real-time reporting phase, the EV uses its assigned pseudonyms π_i and the corresponding session keys for encryption and authentication. Each pseudonym can be used only once to guarantee pseudonym unlinkability. Let m be the plaintext message that the EV wants to report to the utility. Before submitting a report, EV e generates an account field acc_i indicating how the utility should reward the EV (e.g., through anonymous BitCoin transaction [26]). The EV then generates a random secret r_i and commitment $C(acc_i, r_i)$, blinds the commitment using b_e, and sends to the utility

$$\text{msg 7: } \text{EtM}_{k_i^E}^{k_i^A}\{\pi_i, m, b_e(C(acc_i, r_i))\} \tag{9}$$

where the message is first encrypted using k_i^E and then the MAC on the ciphertext is computed using k_i^A. Note that the EV can pre-compute $b_e(C(acc_i, r_i))$ to save time during real-time reporting.

When receiving msg 7, the utility verifies the MAC using k_i^A and decrypts the message using k_i^E. If all verifications succeed, the utility generates $\text{PBS}_{S_\gamma}^{m_0}(b_e(C(acc_i, r_i)))$ by blindly signing $b_e(C(acc_i, r_i))$ with key S_γ, and returns to the EV

$$\text{msg 8: EtM}_{k_i^E}^{k_i^A}\{\text{PBS}_{S_\gamma}^{m_0}(b_e(C(acc_i, r_i)))\} \tag{10}$$

where the message is first encrypted using k_i^E and the MAC on the ciphertext is computed using k_i^A. Note that the blind signature key S_γ used in msg 8 is different than the key S_τ used to generate token τ_i in msg 2.

D. Receipt Submission

In order to get rewarded, the EV must prove to the utility that it has performed useful work (i.e., it has reported) by presenting an authentic receipt to the utility. In particular, after EV e receives msg 8, it computes the receipt

$$\gamma_i = b_e^{-1}(\text{PBS}_{S_\gamma}^{m_0}(b_e(C(acc_i, r_i)))) = \text{PBS}_{S_\gamma}^{m_0}(C(acc_i, r_i)) \tag{11}$$

and verifies that γ_i is a valid signature on $C(acc_i, r_i)$ using public key P_γ. The EV may choose to compute the receipt right after it receives msg 8, or it could delay the computation. After the EV has computed the receipt γ_i, it sends to the utility

$$\text{msg 9: } acc_i, r_i, \gamma_i \tag{12}$$

We do not specify here how the EV should authenticate msg 9. If the EV is willing to reveal its true identity, it can authenticate msg 9 using its digital signatures. Otherwise it may choose some privacy-preserving authentication (e.g., group signature) to authenticate the message and to protect its integrity. Since this message does not need to be sent in real time, the EV may also choose to send it in an anonymous way (e.g., using Tor [18]).

When receiving msg 9, the utility computes $C(acc_i, r_i)$ from acc_i and r_i, and verifies that γ_i is a valid signature on $C(acc_i, r_i)$, and rewards the EV as specified in acc_i.

Note that to defeat timing attacks, after the EV receives msg 8, it should wait for a random amount of time before sending msg 9 to the utility.

V. Security and Privacy Analysis

Anonymity: To protect the EV's anonymity, the utility must not be able to link the EV's true identity e with any of its pseudonyms π_i. Since msg 1 and 2 are signed with digital signatures, the utility can link e to $b_e(C(g^{x_i}, z_i))$. From msg 3-6 the utility learns that $g^{x_i}, \pi_i, C(g^{x_i}, z_i)$ belong to the same EV. However, since the utility cannot link $C(g^{x_i}, z_i)$ from $b_e(C(g^{x_i}, z_i))$, it cannot link pseudonym π_i to EV e.

Location Privacy: Lynx protects the EV's location privacy by allowing the EV to establish anonymous unlinkable pseudonyms. Note that the attacker cannot tell whether two pseudonyms π_i, π_j belong to the same EV. Since each EV uses each of its pseudonyms at most once, the attacker cannot infer the victim EV's trajectory by linking the different pseudonyms used by the victim EV.

Chosen Plaintext Attack: Since the utility does not know the plaintext m before signing the blinded message $b(m)$, the attacker might get the utility's signature on an arbitrarily chosen message. To mitigate chosen plaintext attacks, the utility uses S_τ to blindly sign the authorization token τ and uses another key S_γ to sign the receipt γ, and both S_τ and S_γ are different from the key S_U used to generate signature σ_U in msg 2 and msg 4. In particular, a malicious EV cannot use a token τ_i obtained from msg 2 as a receipt γ_i in msg 9 since they are signed with different keys.

Forging Token/Receipt: In order to forge the authorization token $\tau_i = \text{PBS}_{S_\tau}^{m_0}(C(g^{x_i}, z_i))$ used during key establishment, the attacker would need to forge the utility's signature on $C(g^{x_i}, z_i)$, and would need to provide the corresponding g^{x_i} and z_i. Since the authorization token τ_i is issued only after the utility verifies EV e's digital signature σ_e in msg 1, an outside attacker cannot cause the utility to blindly sign an arbitrarily chosen message. Even if the attacker successfully gets the utility's signature $\text{PBS}_{S_\tau}^{m_0}(m)$ on some random message m (not chosen by the attacker), the attacker cannot construct g^{x_i} and z_i such that $C(g^{x_i}, z_i) = m$ since C is secure and information concealing. Similarly, the attacker cannot forge the receipt γ_i either.

Man-in-the-Middle Attack: Lynx uses an anonymous authenticated version of Diffie-Hellman to prevent man-in-the-middle attacks during the key exchange. Since the utility digitally authenticates its Diffie-Hellman message, the EV authenticates the utility's true identity. By validating the authorization token τ_i, the utility also knows that the EV has been authorized in msg 2.

Replay Attack: All Lynx messages include a fresh timestamp generated by the sender. An outdated message replayed by the attacker can thus be easily recognized. The only message without authentication is the Diffie-Hellman initiation message (msg 3). However, since the attacker does not know the secret z_i in msg 5, replaying msg 3 does not allow the attacker to establish session keys with the utility.

Token Collection Attack: A malicious EV could try to collect tokens through multiple days and use them all at once to establish more than N pseudonyms. Lynx prevents such attacks by using partially blind signatures (PBS) that allows the utility to include the current date m_0 in the generated token τ. Note that the EV cannot change the appended date m_0 in τ. The utility checks that the date in the authorization token τ is the same as the current date. This guarantees the freshness of the token and defeats the token collection attack.

Unknown Key-Share Attack: The Diffie-Hellman key exchange (msg 3-5) in Lynx is similar to the STS protocol [15]. The original STS protocol has been shown to be vulnerable to Unkown Key-Share (UKS) attacks [27], where A mistakenly believes that the shared secret, which is actually shared between honest parties A and B, is shared between A and the attacker C. The UKS attack on the EV would not work since the EV knows the utility's public key certificate before executing the key exchange protocol [27]. The UKS attack on the utility does not achieve anything for the attacker, because by design the utility does not know the identity of the EV.

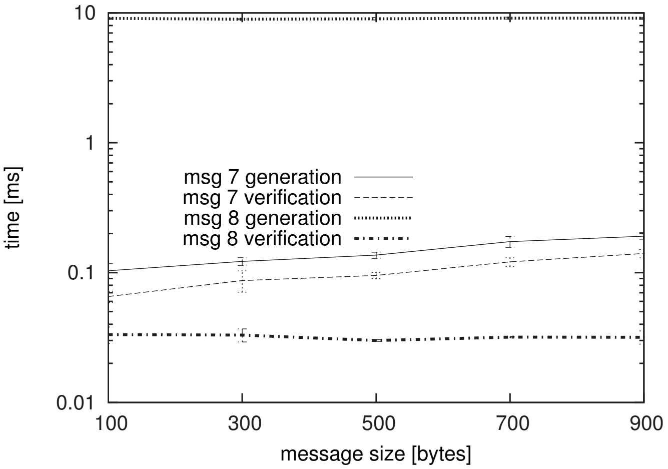

Fig. 3. Generation and verification cost of msg 7 and 8 with different message sizes.

VI. Evaluation

We implemented Lynx in C++ using Crypto++ 5.6.2 library and blind RSA signature. We evaluated our implementation on the Raspberry Pi 2 Model B platform with 900 MHz Quad Core Processor and 1GB RAM. At the time of writing the Pi 2 platform costs $35 (USD).

Since only msgs 7 and 8 need to be exchanged in real time, we focus on the computation complexity of these two messages. In Fig. 3 we show the time required to generate and to verify msg 7 and msg 8. Even With 900 bytes of payload it takes less than 0.2 ms to generate msg 7 because $b_e(C(acc_i, r_i))$ in msg 7 can be pre-computed. The generation of msg 8 is two orders of magnitude more computationally intensive due to blind signature generation, but this message is generated by the utility, and can thus be performed on more powerful computing platforms.

VII. Conclusion

In this paper we proposed Lynx for anonymous authenticated real-time reporting of EV information. Lynx protects EV's privacy by allowing the EV to anonymously establish unlinkable pseudonyms. To minimizes EV's real-time computation overhead, Lynx uses computationally efficient symmetric key-based encryption and authentication for real-time reporting, and performs computationally intensive operations when the EV is parked. Lynx is also reward-compatible, and allows EVs to obtain anonymous receipts for the reports they submit. Our implementation on Raspberry Pi 2 indicates that the computation overhead of generating and verifying real-time reports is less than 10 ms, which makes Lynx practical for real-time reporting.

References

[1] S. Deilami, A. Masoum, P. Moses, and M. Masoum, "Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile," *Smart Grid, IEEE Transactions on*, vol. 2, no. 3, pp. 456–467, Sept 2011.

[2] O. Sundstrom and C. Binding, "Planning electric-drive vehicle charging under constrained grid conditions," in *Power System Technology (POWERCON), 2010 International Conference on*, Oct 2010, pp. 1–6.

[3] O. Ardakanian, S. Keshav, and C. Rosenberg, "Real-time distributed control for smart electric vehicle chargers: From a static to a dynamic study," *Smart Grid, IEEE Transactions on*, vol. 5, no. 5, pp. 2295–2305, Sept 2014.

[4] N. Soltani, S.-J. Kim, and G. Giannakis, "Real-time load elasticity tracking and pricing for electric vehicle charging," *Smart Grid, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[5] H. Qin and W. Zhang, "Charging scheduling with minimal waiting in a network of electric vehicles and charging stations," in *VANET '11*.

[6] E. Sortomme and M. El-Sharkawi, "Optimal scheduling of vehicle-to-grid energy and ancillary services," *Smart Grid, IEEE Transactions on*, vol. 3, no. 1, pp. 351–359, March 2012.

[7] 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.

[8] H. Li, G. Dán, and K. Nahrstedt, "Proactive Key Dissemination-Based Fast Authentication for In-Motion Inductive EV Charging," in *IEEE International Conference on Communications (ICC), 2015*.

[9] ——, "FADEC: Fast authentication for dynamic electric vehicle charging," in *IEEE Conference on Communications and Network Security (CNS), 2013*.

[10] ——, "Portunes: Privacy-preserving fast authentication for dynamic electric vehicle charging," in *IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014*.

[11] Q. Li and G. Cao, "Providing Efficient Privacy-Aware Incentives for Mobile Sensing," in *IEEE ICDCS '14*.

[12] 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*.

[13] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Advances in Cryptology ASIACRYPT 2000*, ser. Lecture Notes in Computer Science, T. Okamoto, Ed. Springer Berlin Heidelberg, 2000, vol. 1976, pp. 531–545.

[14] T. Okamoto, "Efficient blind and partially blind signatures without random oracles," in *Proceedings of the Third Conference on Theory of Cryptography*, ser. TCC'06, 2006, pp. 80–99.

[15] 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.

[16] A. C.-C. Yao and Y. Zhao, "Oake: A new family of implicitly authenticated diffie-hellman protocols," in *ACM CCS '13*.

[17] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: Privacy-aware people-centric sensing," in *MobiSys '08*.

[18] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *USENIX Security '04*.

[19] U. Moeller, L. Cottrell, P. Palfrader, and L. Sassaman, "Mixmaster protocol version 2," *IETF Internet-Draft*.

[20] E. De Cristofaro and C. Soriente, "Extended capabilities for a privacy-enhanced participatory sensing infrastructure (pepsi)," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 12, pp. 2021–2033, Dec 2013.

[21] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "SPPEAR: Security & Privacy-preserving Architecture for Participatory-sensing Applications," in *ACM WiSec '14*.

[22] M. Naor and B. Pinkas, "Oblivious Transfer with Adaptive Queries," in *CRYPTO '99*.

[23] M. Hallenbeck, M. Rice, B. Smith, C. Cornell-Martinez, and J. Wilkinson, "Vehicle volume distributions by classification," 1997.

[24] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," in *ASIACRYPT '96*.

[25] T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO '91*.

[26] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE S&P '13*.

[27] S. Blake-Wilson and A. Menezes, "Unknown key-share attacks on the station-to-station (sts) protocol," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, vol. 1560, pp. 154–170.