PREVENT: A Mechanism for Preventing Message Tampering Attacks in Electric Vehicle Networks

Rohini Poolat Parameswarath
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
rohini.p@nus.edu.sg

Nalam Venkata Abhishek
Infocomm Technology Cluster
Singapore Institute of Technology
Singapore
venkata.abhishek@singaporetech.edu.sg

Biplab Sikdar
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—Electric Vehicle (EV) adoption has been increasing in recent years due to multiple factors. Though EVs offer many advantages, the cyber security of EV networks is often overlooked. When individuals charge their EVs at charging stations, the communication between the EV and the other components of the charging system is through the Internet. It is crucial to understand the potential attacks that an attacker could launch and propose solutions to prevent such attacks to safeguard the EV networks. In this paper, we address message tampering attacks on EV networks and propose a mechanism to prevent them. Existing solutions for message tampering are not suitable for EV networks due to the high computation cost and latency requirements. In the proposed solution, EVs generate authentication parameters based on the charging requests they transmit. The authentication parameters are delivered to a central server together with the charging requests. This enables the server to verify the integrity of the received charging requests. Since the proposed mechanism does not include computationally expensive operations, it does not add significant cost. We present a formal security proof to show that the proposed mechanism provides protection from message tampering attacks and achieves several security properties in the EV charging framework. A performance analysis is also presented to show the computation cost of the proposed mechanism.**

***Index Terms*—Electric Vehicle (EV) networks, integrity, message tampering attacks.**

I. Introduction

There has been an acceleration in the adoption of EVs due to various reasons. It is estimated that, by 2040, out of three vehicles one would be electric [1]. Incentives provided by governments across the globe, increase in traditional fuel prices, and public awareness of the need to use less gasoline and reduce carbon emissions are some of the key factors driving EV adoption. EVs use Grid to Vehicles (G2V) or Vehicle to Grid (V2G) systems for charging. When the EV battery is low, it can be charged from a charging station. When the EV battery has more electricity than required, it can sell the excess electricity to the charging station [2].

EV charging system mainly comprises the charging station management system (CSMS) controlled by various Service Providers (P) and the Charging Stations (C) [3]. The service provider is in charge of generating, transmitting, and distributing electricity to the charging stations. The service provider also has a central server to store information about the users and the charging stations. The charging stations are located in different locations to enable EV users to charge their vehicles while on the move. An EV sends a charging request to the charging station and the charging station forwards it to the central server. The Open Charging Point Protocol (OCPP) is widely used for communication between the CSMS and charging stations. OCPP works with any EV charging protocol. Since it is an open protocol, new security features can be easily added to it.

With the adoption of EVs, cyber security concerns about the EV network, particularly about the charging infrastructure, have also been emerging. Due to the rising interest in EVs, security concerns about the charging infrastructure have to be addressed [4]–[6]. Since the communication among the service providers, the charging stations, and the electric vehicles is through an insecure medium, the Internet, an attacker may launch multiple attacks against the electric vehicle network. By compromising a charging station, an attacker may eavesdrop on the charging requests sent from multiple EVs and modify them which will affect the integrity of the data being sent [7]. In this paper, we address such message tampering attacks on EV networks. Multiple solutions have been employed to protect the integrity of the messages in different domains. Message tampering attacks can be defended to an extent through various cryptographic techniques. Another common method employed to ensure data integrity is through the use of the message authentication code (MAC). In EV charging systems, the computation and communication costs should be minimal. Therefore, a lightweight technique that minimizes latency and preserves data integrity is required for EV charging systems. To address this problem, we propose a lightweight mechanism to protect data integrity in EV charging systems in this paper.

A. Related Work

There have been many studies on the security of EV networks. EVs are susceptible to all attacks that can be made against other vehicles and on top of that, the EV charging systems are susceptible to additional attacks [8]. The authors of [9] discussed cyber attacks against EV charging infrastructure and the available security solutions to protect

it from such attacks. The authors of [10] proposed a model that helps to isolate malware-infected nodes in the electric vehicle infrastructure. The vulnerabilities that exist in the EV charging process are presented in [11]. The cyber security issues faced by the EV charging stations and solutions to mitigate their impacts are presented in [12]. Details such as EV identification, location of the EV, energy tariff, meter reading, and control commands are provided by the EV user while charging the EV. An attacker may tamper with such messages for free charging [13]. Further, the authors of [13] recommend using digital signatures for all messages exchanged between the EVs and the charging stations or using encryption techniques to prevent such message alteration attacks. However, using such cryptographic techniques for all messages is computationally expensive. A protocol for V2G that protects the EV network from several attacks including message tampering attacks was proposed in [14]. However, the protocol in [14] requires additional hardware, the Physical Unclonable Function (PUF). Hence, we propose a computationally lightweight mechanism, that does not require any additional hardware, called PREVENT to detect message tampering attacks in EV networks.

II. System and Adversary Model

In this section, we briefly introduce the system and the adversary models.

A. System Model

We consider the system model shown in Figure 1. The electric vehicles are represented as V_i, for $i \in \{1, 2, \cdots\}$. They charge from one of the charging stations denoted as C_j, for $j \in \{1, 2, \cdots\}$. The service provider is responsible for transmitting and distributing electricity to the charging stations. Electric vehicles, charging stations, and service providers communicate over the Internet. The vehicles transmit charging requests to the nearby charging station when they need to charge.

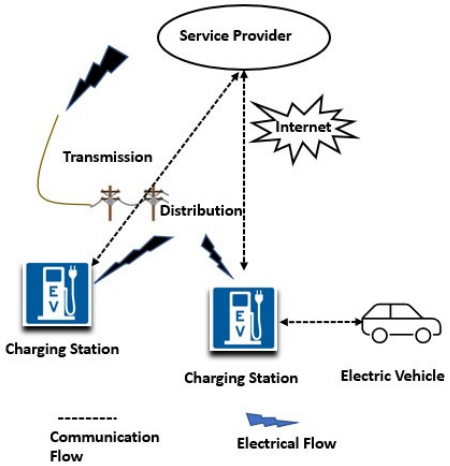

Fig. 1: System model.

B. Adversary Model

Electric vehicles, charging stations, and service providers communicate over the Internet which is an insecure medium. Consider the scenario where a number of EVs $\mathcal{EV} = EV_1, EV_2, \cdots, EV_n$ send charging requests to the service provider $\mathcal{S}$ through a charging station $\mathcal{C}$. Suppose an attacker $\mathcal{A}$ compromised $\mathcal{C}$ such that $\mathcal{A}$ can listen, replay, inject, and modify the charging requests sent by the EVs. Using the following queries, we can model the attacker's capabilities:

- `Monitor` models $\mathcal{A}$'s ability to monitor the messages exchanged between the EV and $\mathcal{C}$.
- `Capture` models $\mathcal{A}$'s ability to capture the messages exchanged between the EV and $\mathcal{C}$.
- `Send(m)` is used to model $\mathcal{A}$'s ability to impersonate an EV and send a message `m` to $\mathcal{C}$.

$\mathcal{A}$ may call the `Monitor`, `Capture`, and `Send` queries a polynomial number of times.

III. PREVENT: Proposed Solution

A. High-level Workflow

We consider a vehicular network with multiple electrical vehicles moving around. The vehicles are represented using V_i, for $i \in \{1, 2, \cdots\}$. The area is also equipped with charging stations denoted using C_j, for $j \in \{1, 2, \cdots\}$. We now demonstrate the algorithm for preventing message tampering attacks launched by an adversary who has compromised the charging station C_j. The high-level workflow of the proposed solution is shown in Figure 2 with the following steps:

1) A registered electric vehicle generates the charging request details to send to the charging station. This part of the message is not encrypted.
2) The electric vehicle calculates the validation parameter for the charging request details. Then, it encrypts the validation parameter with the secret key k_i.
3) The electric vehicle concatenates the charging request details and the encrypted validation parameter and sends it to the charging station.
4) The charging station forwards the charging request details and the encrypted validation parameter received from the electric vehicle to the service provider.
5) The service provider calculates the validation parameter. Then, it decrypts the received encrypted validation parameter with k_i. After that, it verifies the received validation parameter by comparing it with the calculated parameter. If no message tampering has taken place, then the charging request is not modified, and both the received and calculated validation parameters will be the same.
6) If the validation parameter verification is successful, the charging request is granted.

The advantage of using PREVENT is that, even if a small percentage of vehicles transmit the validation parameter, message tampering can be detected with a high probability.

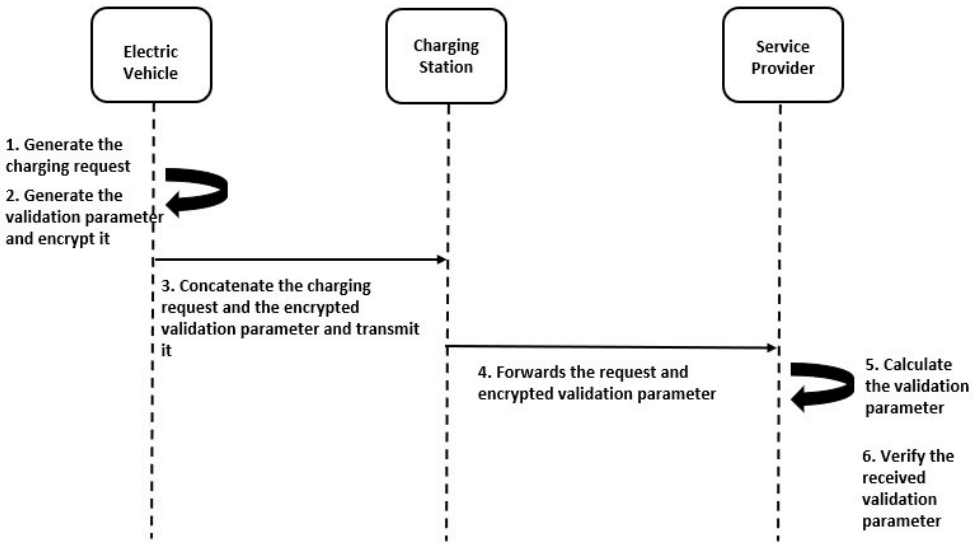

Fig. 2: The high-level workflow of the solution.

B. Detailed Protocol

The algorithm is divided into the following three phases:

1) Registration Phase: The electric vehicles register with the central server in this phase. Then, the EVs and the central server exchange a key required for transmitting the charging request. Note that this phase needs to be executed only once. A key and a pseudo-random sequence are shared between the server and the EV at the end of the registration phase.
2) Message Transmission Phase: In this phase, the electric vehicle transmits the charging request R_n for $n \in \{1, 2, \cdots\}$ to the central server S via the charging station C_j. The subscript of R_n also denotes the order in which the charging requests are transmitted i.e. the order of transmission is R_1 followed by R_2 followed by R_3 and so on.
3) Verification Phase: In this phase, the server verifies the validity of the charging request and sends approval to the electric vehicle via the charging station if the received message is not tampered with by an adversary.

The three phases are discussed in detail below:

1) Registration Phase: This phase is executed only once. In this phase, the EVs register with the central server with their identities. A key is shared between the central server and the EV at the end of this phase. The key exchanged between the server and vehicle V_i is denoted using k_i for $i \in \{1, 2, \cdots\}$. In addition to the shared key, the electric vehicle and the central server share a pseudo-random sequence denoted using S_0. The key k_i and the pseudo-random sequence S_0 will be used while transmitting the charging request in the next phase.

2) Message Transmission Phase: Let us say the vehicle V_i is in the vicinity of the charging station C_j and transmit the corresponding charging request to C_j to charge the vehicle. V_i generates a validation parameter, denoted using P_n, once a charging request is initiated. This implies that the n^{th} charging request, denoted using R_n, has two parts: Q_n and E_n. E_n is the encrypted version of P_n. The base-2 representation of the first part of the charging request is represented using Q_n. This part includes information about the charging requirements and requested time. The base-2 representation of the second part is represented using P_n. P_n is generated by using the Q_i part of the past N charging requests, i.e., by using $\{\mathsf{Q}_{n-N}, \cdots, \mathsf{Q}_{n-1}\}$ as follows:

$$\mathsf{P}_n = \texttt{NAND}(\mathsf{Q}_{n-N}, \mathsf{Q}_{n-N+1}, \cdots, \mathsf{Q}_{n-1}, S_n). \tag{1}$$

where NAND refers to the bit-wise NAND operation and S_n is a pseudo random sequence shared between the vehicle and the server. If the values of n is less than or equal to the value of N, then P_n is generated as follows:

$$\mathsf{P}_n = \texttt{NAND}(\mathsf{Q}_1, \mathsf{Q}_2, \cdots, \mathsf{Q}_{n-1}, S_n). \tag{2}$$

The value of S_n is generated as follows:

$$S_n = H(S_{n-1}) \tag{3}$$

where $H(\cdot)$ denotes the hash function. Note that S_n has to be zero-padded to ensure that its length is the same as that of Q_n. The second part, i.e., the P_n is encrypted before transmission. We denote the encrypted version using E_n and is defined as follows:

$$\mathsf{E}_n = [\mathsf{P}_n]_{k_i} \tag{4}$$

where E_n is the base-2 representation of E_n. The charging request message to be transmitted, i.e., R_n is generated by concatenating Q_n and E_n. Then, the EV transmits R_n to the charging station. Upon receiving R_n from the EV, the charging station forwards it to the service provider.

3) Verification Phase: In this phase, the central server verifies the charging request from the EV. The request received from the EV is denoted using a different notation $\hat{R}_n$. If an adversary has not tampered with the charging request, R_n is equal to $\hat{R}_n$. To verify that the charging request message is not tampered with, the server extracts the second part of the charging request message, $\hat{E}_n$. We can assume that the charging requests received prior to $\hat{R}_n$ are verified by the server. Therefore it would be possible for the server to generate E_n similar to Equation 4 as it has access to the previous N charging requests. The server confirms that the charging request is not modified by an adversary iff E_n is equal to $\hat{E}_n$. Else, the server rejects the charging request from the EV.

IV. Formal Security Proofs for the Proposed Protocol

Lemma 1. *The key k_i cannot be predicted.*

Proof. The key k_i is a random number. The only possibility to predict it is to make a random guess. If there are n bits in k_i, the adversary's advantage in predicting k_i, $\boldsymbol{\alpha}_{\mathcal{A}}^{k_i} = \frac{1}{2^n}$. Further, $\boldsymbol{\alpha}_{\mathcal{A}}^{k_i}$ reduces as the number of bits in k_i increases. Hence, the probability of predicting the key by the adversary is negligible. □

Lemma 2. *The pseudo-random sequence S_i cannot be predicted.*

Proof. The pseudo-random sequence S_i is a random number. The only possibility to predict it is to make a random guess. If there are n bits in S_i, the adversary's advantage in predicting S_i, $\boldsymbol{\alpha}_{\mathcal{A}}^{S} = \frac{1}{2^n}$. Further, $\boldsymbol{\alpha}_{\mathcal{A}}^{S}$ reduces as the number of bits in S_i increases. Hence, the probability of predicting the pseudo-random sequence by the adversary is negligible. □

Theorem 1. *Replay Attacks: The attacker cannot replay previous messages from the EV to the charging station.*

Proof. The replay attack where an adversary $\mathcal{A}$ replays previously transmitted messages from the EV can be modelled through the following game:

1) An EV EV_1 initiates message transmission to the charging station $\mathcal{C}$.
2) $\mathcal{A}$ executes the query `Monitor` a polynomial number of times.
3) $\mathcal{A}$ executes the query `Capture` a polynomial number of times.
4) $\mathcal{A}$ executes the query `Send` to send the captured message to $\mathcal{C}$. $\mathcal{C}$ forwards the message to the service provider.
5) $\mathcal{A}$ wins the game if the validation parameter is verified successfully by the service provider.

When the service provider verifies the validation parameter, since E_n is not equal to $\hat{E}_n$, the verification will fail. As a result, $\mathcal{A}$ cannot send previous messages successfully. Therefore, the proposed mechanism is robust against replay attacks. □

Theorem 2. *Message Tampering Attacks: The messages exchanged between the EV and the charging station cannot be modified by an adversary.*

Proof. The message tampering attack where an adversary $\mathcal{A}$ modifies the transmitted messages between the EV and the charging station can be modelled through the following game:

1) An EV EV_1 initiates message transmission to the charging station C.
2) $\mathcal{A}$ executes the query `Monitor` a polynomial number of times.
3) $\mathcal{A}$ executes the query `Capture` a polynomial number of times and modifies the messages.
4) $\mathcal{A}$ executes the query `Send` to send the modified messages to $\mathcal{C}$. $\mathcal{C}$ forwards the message to the service provider.
5) $\mathcal{A}$ wins the game if the validation parameter is verified successfully by the service provider.

P_n is encrypted with k_i to get E_n before transmission at EV_1. $\mathcal{A}$ needs to know k_i to decrypt E_n. From Lemma 1, $\mathcal{A}$'s advantage in predicting k_i is negligible. Hence, $\mathcal{A}$ cannot decrypt and modify the messages. Therefore, the proposed mechanism is secure against message tampering attacks. □

V. Performance Analysis

In this section, we analyze the computation cost incurred at the electric vehicle and at the service provider to confirm whether the proposed mechanism is a computationally reasonable solution.

A. Computation Cost at the Electric Vehicle

The validation parameter is computed using Equation 1. The pseudo-random sequence is calculated using Equation 3 by using hash operation. If there are N previous charging requests, $NAND$ operation is applied to them, and the pseudo-random sequence to find the validation parameter. Hence, there are N $NAND$ operations. Finally, the validation parameter is encrypted with the shared symmetric key k_i. Let us denote the time taken by hash, NAND, and encryption operations as T_H, T_{NAND}, and T_{Enc}, respectively. Hence, the total computation cost is $T_H + N \times T_{NAND} + T_{Enc}$. We simulated the experiments on a personal computer with an Intel Core i5 CPU, 3.20 GHz clock, and 8 GB of RAM. The time taken by hash, NAND, and encryption operations is 0.11 ms, 0.01 ms, and 0.17 ms, respectively. The computation cost incurred at the EV when the number of charging requests increases from 5 to 50 is shown in Figure 3.

B. Computation Cost at the Service Provider

Next, we analyze the computation cost incurred at the service provider. The service provider calculates the validation parameter for every vehicle that sends a charging request. Suppose an electric vehicle has sent N charging requests to the charging stations so far. When the vehicle sends the next charging request, the service provider computes the validation parameter using Equation 1. The computation cost to calculate

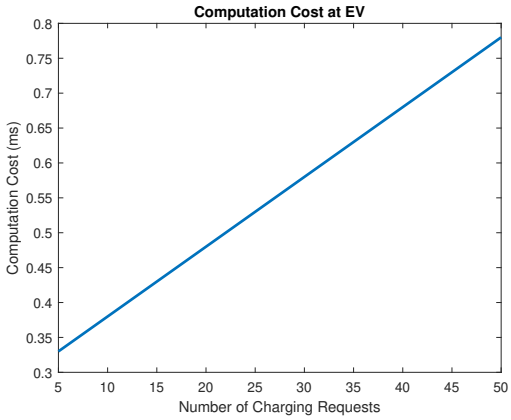

Fig. 3: Computation cost at the EV as the number of charging requests increases.

the validation parameter is $T_H + N \times T_{NAND} + T_{Enc}$. If there are l vehicles and the service provider calculates the validation parameter for every vehicle, the computation cost at the service provider is $l \times (T_H + N \times T_{NAND} + T_{Enc})$. We calculated the computation cost to generate the validation parameter at the service provider when the number of electric vehicles increases from 5 to 30 for $N = 10$. The graph is shown in Figure 4. From the analysis and figures, it is very clear that the computation cost of the proposed mechanism is reasonable.

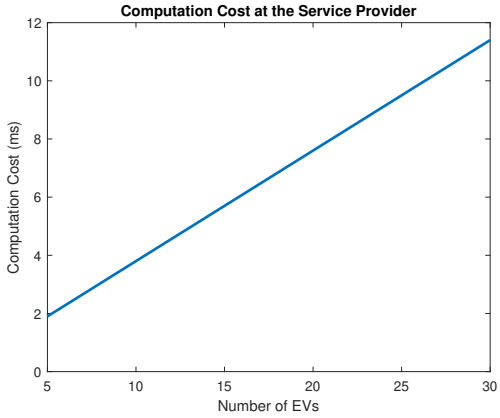

Fig. 4: Computation cost at the service provider as the number of EVs increases.

VI. Conclusion and Future Work

We proposed a lightweight mechanism to prevent message tampering attacks in electric vehicle networks. The electric vehicles calculate validation parameters periodically with lightweight cryptographic operations. They send charging requests and encrypted validation parameters to the charging stations which are forwarded to the service provider. These validation parameters are compared by the service provider with the computed parameters to detect the presence of an adversary who modifies the charging request messages. Our analysis shows that the proposed mechanism provides protection from message tampering and other attacks and the computation cost is reasonable.

In this paper, we considered an adversary who tampers with charging requests from every vehicle. As a part of our future work, we would consider an attacker strategy where the adversary selectively tampers the charging requests, i.e., the adversary does not tamper with every vehicle's charging requests. We will also consider a situation where the vehicles transmit incorrect validation parameters.

VII. Acknowledgement

This paper was supported in part by the Ministry of Education, Singapore under grants R-263-000-E78-114 and R-263-001-E78-114.

References

[1] "Open vs. Closed charging stations: Advantages and disadvantages, Los Angeles, CA, USA, 2018," 2018. [Online]. Available: https://greenlots.com/wp-content/uploads/2018/10/Open-Standards-White_Paper-compressed.pdf

[2] I. Sami, Z. Ullah, K. Salman, I. Hussain, S. M. Ali, B. Khan, C. A. Mehmood, and U. Farid, "A bidirectional interactive electric vehicles operation modes: Vehicle-to-grid (v2g) and grid-to-vehicle (g2v) variations within smart grid," in *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, 2019, pp. 1–6.

[3] M. J. F. Buve and P. Klapwijk, "OCPP 2.0.1 part 1—Architecture topology," 2020. [Online]. Available: https://www.openchargealliance.org/protocols/ocpp-201/

[4] H. ElHussini, C. Assi, B. Moussa, R. Atallah, and A. Ghrayeb, "A tale of two entities: Contextualizing the security of electric vehicle charging stations on the power grid," *ACM Transactions on Internet of Things*, vol. 2, no. 2, pp. 1–21, 2021.

[5] Z. Garofalaki, D. Kosmanos, S. Moschoyiannis, D. Kallergis, and C. Douligeris, "Electric vehicle charging: A survey on the security issues and challenges of the open charge point protocol (ocpp)," *IEEE Communications Surveys Tutorials*, vol. 24, no. 3, pp. 1504–1533, 2022.

[6] R. Gottumukkala, R. Merchant, A. Tauzin, K. Leon, A. Roche, and P. Darby, "Cyber-physical system security of vehicle charging stations," in *2019 IEEE Green Technologies Conference(GreenTech)*, 2019, pp. 1–5.

[7] C. Alcaraz, J. Lopez, and S. Wolthusen, "Ocpp protocol: Security threats and challenges," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2452–2459, 2017.

[8] A. Verma, R. Saha, G. Kumar, and T.-h. Kim, "The security perspectives of vehicular networks: a taxonomical analysis of attacks and solutions," *Applied Sciences*, vol. 11, no. 10, p. 4682, 2021.

[9] J. Antoun, M. E. Kabir, B. Moussa, R. Atallah, and C. Assi, "A detailed security assessment of the ev charging ecosystem," *IEEE Network*, vol. 34, no. 3, pp. 200–207, 2020.

[10] S. Mousavian, M. Erol-Kantarci, and T. Ortmeyer, "Cyber attack protection for a resilient electric vehicle infrastructure," in *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2015, pp. 1–6.

[11] O. G. M. Khan, E. El-Saadany, A. Youssef, and M. Shaaban, "Impact of electric vehicles botnets on the power grid," in *2019 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, 2019, pp. 1–5.

[12] Z. Pourmirza and S. Walker, "Electric vehicle charging station: Cyber security challenges and perspective," in *2021 IEEE 9th International Conference on Smart Energy Grid Engineering (SEGE)*, 2021, pp. 111–116.

[13] C. Hodge, K. Hauck, S. Gupta, and J. C. Bennett, "Vehicle cybersecurity threats and mitigation approaches," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2019.

[14] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for v2g using physical unclonable function," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7234–7246, 2020.