Lightweight Mutual Authentication Protocol for V2G Using Physical Unclonable Function

Gaurang Bansal, *Member, IEEE*, Naren, Vinay Chamola, *Member, IEEE*, Biplab Sikdar, *Senior Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE* and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Electric vehicles (EVs) have been slowly replacing conventional fuel based vehicles since the last decade. EVs are not only environment-friendly but when used in conjunction with a smart grid, also open up new possibilities and a Vehicle-Smart Grid ecosystem, commonly called V2G can be achieved. This would not only encourage people to switch to environment-friendly EVs or Plug-in Hybrid Electric Vehicles (PHEVs), but also positively aid in load management on the power grid, and present new economic benefits to all the entities involved in such an ecosystem. Nonetheless, privacy and security remains a serious concern of smart grids. The devices used in V2G are tiny, inexpensive, and resource constrained, which renders them susceptible to multiple attacks. Any protocol designed for V2G systems must be secure, lightweight, and must protect the privacy of the vehicle owner. Since EVs and charging stations are generally not guarded by people, physical security is also a must. To tackle these issues, we propose Physical Unclonable Functions (PUF) based Secure User Key-Exchange Authentication (SUKA) protocol for V2G systems. The proposed protocol uses PUFs to achieve a two-step mutual authentication between an EV and the Grid Server. It is lightweight, secure, and privacy preserving. Simulations show that the proposed protocol performs better and provides more security features than state-of-the-art V2G authentication protocols. The security of the proposed protocol is shown using a formal security model and analysis.**

***Index Terms*—authentication, security, smart grid, V2G, PUF**

I. INTRODUCTION

Electric Vehicles' batteries enable the functionality of V2G networks. The purpose of V2G is to manage the energy trading for battery-poweblack electric vehicles as well as the power grid. This is necessary in order to use the grid's energy more efficiently [1]. The electrical energy stored in the EV batteries can serve as a source for the power grid and other energy deficient EVs. When the load on the grid is high, the energy stored in the EVs' batteries could be used to pump power into the grid. On the other hand, when the load on the grid is low, the excess power in the grid could be used to charge the EV batteries and avoid wastage [2]. V2G networks could also be used for power regulation [3] or for storing power generated by renewable sources such as wind power [4]. Thus, nowadays, V2G for smart grids presents great practical applications.

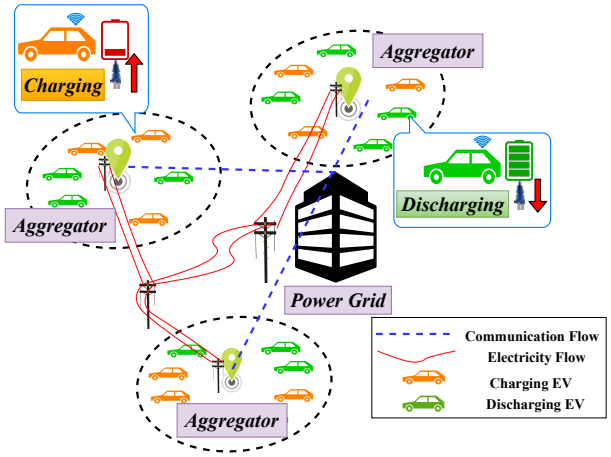


Fig. 1: System model

The global demand for electrical power is predicted to climb 82% by the year 2030. Therefore, Power grids are aiming to reduce the number of additional generators required. They employ demand-response techniques [5] to reduce power consumption and increase efficiency. Although such techniques offer many benefits, security and privacy issues remain to be significant downsides [6, 7]. A lot of information is communicated during energy exchange between a vehicle and a service provider. However, an adversary could compromise this flow of information, either by tampering with it or capturing it entirely [8]. This could lead to unfair or imbalanced energy transactions between the two parties. Moreover, the victim's information (that could be captured) may be used in criminal activities and targeted advertisements. The devices used in V2G are inexpensive, small, and simple [9]. The EVs are usually parked in locations which are easy to access. This means that an adversary could easily capture the V2G devices on these vehicles. Therefore, it is important to make V2G entities/devices secure against physical attacks. For instance, an adversary could access security keys stored in the device memory and initiate various attacks. Physical Unclonable Functions (PUFs) have emerged as a promising solution for protection against physical attacks. PUFs eliminate the need to store secret keys in the devices' memory and rely on the exchange of challenge-response pairs. The challenge-response mechanism of PUFs exploits the inherent fabrication or manufacturing process variabilities involved in making integrated

Gaurang Bansal, Naren and Vinay Chamola are with the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus, India 333031 (e-mail: h20140128@pilani.bits-pilani.ac.in, f2015547@pilani.bits-pilani.ac.in, vinay.chamola@pilani.bits-pilani.ac.in).

Biplab Sikdar is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: bsikdar@nus.edu.sg).

Neeraj Kumar is with the Department of Computer Science, Thapar University, Patiala, India 147004 (e-mail: neeraj.kumar@thapar.edu).

Mohsen Guizani is with the Department of Computer Science, Qatar University, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier: XXXXXXXXXXXX

circuits (ICs) [10]. The response or output of a PUF depends on both the input as well as the physical microstructure of the device [11]. The physical randomness induced through the fabrication process variations makes each PUF device unique, i.e., two identical copies can never be made.

In V2G systems, an aggregator is a charging station which acts as a mediator between the EVs and the power grid. Secure communications in such scenarios require authentication between the EVs and the aggregator, between aggregator and the grid, and between EVs and the grid as well. The proposed Secure User Key-Exchange Authentication (SUKA) protocol achieves the authentications mentioned above using a two-stage process. Using pseudonym IDs (PID), the identity of the vehicle is masked to protect the vehicle owner's identity and location. Two different session keys are established in SUKA, one between the aggregator and the grid, and another between any EV and the aggregator. These session keys are a function of the PUFs installed on the aggregator and the EV, respectively. This ensures the secrecy of the communication and eliminates the need to store any secret keys in the memories of the EVs and the aggregator. The proposed protocol uses simple cryptographic operations, which makes it lightweight and energy efficient. The number of message exchanges is also limited, which results in a lower communication overhead. The major contributions of this paper are highlighted below:

- We propose a security scheme, SUKA, for V2G applications where both the mobile EVs and the stationary aggregators are provisioned with PUFs for secure communication.
- SUKA puts the safety of the EV and its owner as top priority by first achieving authenication of the aggregator with the power grid server. Only if this is achieved, the aggregator will be able to accommodate the EV. Our scheme ensures security even in the case of a compromised aggregator.
- SUKA ensures mutual authentication, identity protection, message integrity and is tolerant to man-in-the-middle (MITM) attacks, impersonation attacks, replay attacks and node tampering attacks without having to store any secret keys in the EVs or the stationary aggregators. The records of EVs cannot be tracked even if the EVs and/or the stationary aggregators are compromised.
- SUKA establishes different session keys between the EV and the aggregator, and between aggregator and the power grid server. These session keys change randomly in each round of authentication and cannot be accessed by an adversary even if it gains physical access to both the EV and the aggregator.

The rest of this paper is organized as follows. Section II discusses the related work in V2G systems. Section III presents a brief introduction to PUFs, the network model, security goals, assumptions for the V2G system, and the notations used in our paper. Section IV presents our MA protocol (SUKA). Section V presents a formal security analysis of the proposed protocol. We analyze the performance of our protocol and compare it with state-of-the protocols in Section VI and finally conclude the paper in Section VII.

II. Related Work

Kempton and Tomic [12] first conceived the idea of V2G in 2004. Before the protocols for V2G networks could be developed, the structure of a V2G network had to be well defined, and the impact of V2G on the power grid had to be analyzed. This work was carried out by the authors in [13, 14, 15, 16, 17]. Privacy, secure communication, and efficiency are among the most important aspects of a V2G protocol [18]. Privacy preservation in V2G environments has received considerable attention in the existing literature [19, 20, 21, 22]. Yang *et al*. have presented a protocol P^2 in [19] which achieves privacy for individual electric vehicles (EVs) and the rewarding scheme which is crucial for proper implementation of V2G. Liu *et al*. present their scheme, $AP3A$, which is capable of identifying whether an EV is in its home or visiting network [20]. $AP3A$ communicates the aggregated power status of the vehicles connected to an aggregator instead of revealing individual power status, thus achieving privacy for each EV. Liu *et al*. have presented a scheme which identifies the different roles played by an individual EV, i.e., customer, storage or generator [22]. In each role, their scheme $ROPS$ addresses different privacy concerns. Tsai and Lo achieve mutual authentication and identity protection with the use of one private key which is given by a third-party anchor. This enables the smart-meters to quickly authenticate with the service provider. Abdallah and Shen propose a computationally less intensive privacy-preserving scheme in [24]. They identify that the authentication of EVs in the V2G system is specifically problematic. Therefore, the power grid takes the responsibility of ensuring the confidentiality and integrity of the communication. By reducing the number of exchanged messages, they achieve less overhead. Odelu *et al*. present a secure authenticated key agreement scheme [25] under the Canett-Krawczyk adversary (CK-adversary) model for smart grids. Shen *et al*. propose a privacy-preserving key agreement protocol for V2G networks in [26]. Their protocol ensures security by the use of a session key and ensures privacy using a self-synchronization mechanism.

Protocols for authentication in V2G environments have been proposed in [27, 28, 29]. Saxena and Choi have presented an authentication scheme for large V2G networks where vehicles move from their home network to other networks as visitors [30]. They propose a mutual authentication scheme which protects against impersonation, key-based and data-based attacks. Tao *et al*. have presented capacity-aware protocol $AccessAuth$ in [31] which takes into consideration the capacity limitations of each V2G network domain, of the EVs, and the mobility of the EVs for admission control. Based on prior information of trust between V2G network domains, they present a high-level authentication model and procedure to ensure that only authorized entities conduct the sessions. Gope and Sikdar have used one-way noncollision hash functions to propose a lightweight mutual authentication protocol [32]. Fouda *et al*. have proposed a lightweight message authentication scheme in [33]. In their scheme, smart meters at different levels in the smart-grid achieve mutual authentication among themselves, and a shared session key is established. They achieve lightweight

message authentication using this shared session key along with a hash-based authentication code mechanism. Although this scheme was presented for smart grid communications, it can very well be extended to V2G networks.

While many privacy-preserving, lightweight mutual authentication, and key establishment protocols exist for V2G systems, none of them provide all the required security and privacy features along with protection against all types of attacks, especially protection against physical attacks. If a protocol does provide perfect security, then it either requires resource-heavy hardware or is computationally complex.

III. Notations

Table I lists the notations used in this paper and their descriptions.

TABLE I: Notations

Notation	Description
V, ID_V	Vehicle and its ID
M, ID_M	Aggregator(mediator) and its ID
G	Grid Server
$\|$	Concatenation operator
$\oplus$	XOR operation
F	A public non-linear function
$\{Msg\}_k$	Message Msg is encrypted using key k
Msg_{P2Q}	Message Msg is sent from V2G entity P to Q
$MAC(X)$	Message authentication code (MAC) of X
N_A, N_B, N_C N_I, N_O, N_V	Nonces generated at different stages
$(C, K), (C', K')$ $(C'', K''), (C^{\#}, K^{\#})$	Challenge-response pairs of PUF

IV. Preliminary Background

A PUF is based on a unique physical property of a device. It is similar to and as unique as the biometrics of a human. The distinguishing attribute of a PUF is that it relies on a physical basis, making it impossible to reproduce a PUF using cryptographic primitives. Additionally, the term "physical unclonable" indicates that it is computationally infeasible or difficult to produce a physically identical PUF [34]. By using PUFs in an interconnected system such as IoT or V2G systems, every single device can have its own unique "fingerprint" which cannot be cloned or reproduced [35]. A PUF behaves like a mathematical function whose input (challenge) and output (response) are both in the form of a string of bits. A PUF function can be represented as:

$$K = PUF(C) \tag{1}$$

where the challenge C is given as input and response, K is the corresponding output to that challenge.

PUFs are designed deliberately so that the response to a particular challenge depends on the individual physical disorder present in the PUF. Therefore, each PUF response is not only a function of the challenge applied, but also a function of its physical disorder. While it is clear that different challenges to the same PUF will give different responses, PUFs also show the following unique characteristics with respect to their input C and output K:

1) If an input C is given to the same PUF many times, it produces the same response K with a very high likelihood.
2) If the same input C is given to different PUFs, the responses obtained from each PUF differ greatly from each other with a very high likelihood.

The characteristics mentioned above apply to non-ideal PUFs. Due to environmental and circuit noises, a non-ideal PUF cannot guarantee the exact same response K for an input C, hence the word 'high likelihood' was used above. There will always be some bit-errors depending on the type of PUF used. Such a flaw would compromise the availability of any system. Although error correcting techniques such as fuzzy extractors could be used in order to combat this problem, that would result in unnecessary overhead for the MA process. Therefore, the PUFs employed in the proposed protocol have to be ideal in nature, i.e., without any bit errors. This would ensure 100% availability of the V2G system.

In the past few years, several types of ideal PUFs have been developed, which ensure 0% Bit-Error-Rate (BER) over a wide range of temperature and voltage fluctuations. The authors of [36] have been able to achieve 0% BER in SRAM PUFs with a low-cost design modification to existing SRAM designs. By using a suitable algorithm to identify strong or consistent cells within the SRAM and by using only those cells for generating Challenge-response pairs, their PUF design is able to withstand any amount of realistic circuit noises. Another extremely promising 0% Bit-Error-Rate (BER) design is the VIA-PUF developed by the authors of [37]. By setting new design rules for regulating the via hole size during IC manufacturing, they have been able to achieve 0% Bit-Error-Rate (BER) for 1000 measurements on 119 chips fabricated using 0.18 μm CMOS technology. Further, no bit-errors were observed in 1000 measurements after a stress test at 125° for 96 hours. These ICs are very small, measuring just a few millimeters in dimensions and require just 1.8 Volts to operate, which makes them ideal for the V2G scenario. Several other works [38, 39, 40] have also been able to achieve 0% Bit-Error-Rate (BER).

These ICs are very small, measuring just a few millimeters in dimensions and require very small voltages (1.8 Volts for VIA-PUF [37]) to operate and hence provide an optimal choice for security provisioning in V2G systems. By using Ideal PUFs with on-board computers in V2G systems, stable key generation can be achieved without the need for any dedicated error correction hardware or software components and thus promise lightweight and high-security performance when used in V2G systems. However, ideal PUFs have been developed very recently and have only been fabricated for research purposes. Techniques to incorporate these PUFs on the on-board computers of V2G systems, or System-on-Chip

designs (SoCs) with built-in PUFs do not exist at present. Therefore, significant work is required to properly implant PUFs in the on-board computers of EVs or aggregators. Research and development of such ideal-PUF based solutions for V2G systems can be considered as future work, but further discussion on this topic is beyond the scope of this paper.

V. System Model

A. Network Model

Figure 1 depicts the system model. This model consists of three entities: EVs, aggregators (or mediators), and the grid. An aggregator is a charging/discharging station where many vehicles can come to charge/discharge their batteries. It acts as a mediator between the EVs and the grid. EVs and aggregators have limited resources, while the grid has significantly larger resources. Aggregators and EVs have similar capabilities, but aggregators have slightly larger memory and computation power. As can be seen in Figure 1, multiple vehicles are connected to an aggregator, and multiple aggregators connect to the power grid. Our objective is to develop a mutual authentication (MA) protocol between EVs and the grid. The device on every vehicle and aggregator is equipped with a PUF. Since a vehicle does not communicate directly with the grid, to achieve MA between these two non-communicating parties, all the intermediary nodes must be authenticated. Thus, MA between the grid and a vehicle can be divided as MA between the aggregator and the grid along with MA between the vehicle and the aggregator. We assume here that there is no shared key between a vehicle and its corresponding aggregator or between an aggregator and the grid. Whenever a new vehicle wants to register on the network, a set of its challenge-response pairs are stored in the grid server. The grid is the only trusted authority, and therefore, challenge-response pairs for all vehicles are stored only in the grid. Nothing else is assumed in further communication.

The server on the power grid starts with a set of initial challenge-response pairs, (C, K) for each aggregator and (C'', K'') for each EV. The aggregator acquires this initial set (C'', K'') for the EV once the aggregator itself mutually authenticated with the grid server. To set up a new aggregator in a location or to deploy a new vehicle or on the roads, initialization involves the initial set $(C'', K'')/(C, K)$ to be sent to the power grid server using a secure channel. This initialization can be done using a time-based one-time password algorithm (TOTP) [41] and an operator using a password. After this exchange, the aggregator or vehicle can function on its own without needing any operator or secure channel. The grid server stores the actual identity ID_M/ID_V, and their corresponding challenge response pairs, (C, K) and $(C", K")$, for each aggregator and vehicle, while the aggregators or vehicles themselves do not store anything. At the end of the protocol, the ID of the EV, ID_V, is replaced with pseudo-identities for further exchanges.

We assume that an adversary can get hold of any communication that is happening between the EV and the aggregator or the aggregator and the grid. An adversary has the power to change, manipulate, and hide the data. It can inject new packets, store the old messages, initiate a session, or pretend to be a valid device. The objective of an attacker or adversary is to gain access to the grid without being noticed. Adversaries may be EV owners who want to exploit the V2G system in order to cheat the service provider to charge their vehicles for free or to get more money from the service provider when they supply power to the grid from their EV. They may also be rogue or unauthorized aggregators set up to cheat EV owners by charging very high prices or not paying the EV owner for the power they acquire. Such aggregators may also be selling personal information of the EV owner to third-party entities for the purpose advertisements. Adversaries may also be criminals who may want to track the location/behaviour of an EV owner by recording the aggregators visited by the EV or criminals who may want to authenticate with the aggregator with some other EV's identity in order to escape the payment.

If an unauthorized or potentially dangerous entity manages to authenticate with the grid server, it may disrupt energy transactions and cause economic damage. Therefore, this paper proposes a MA protocol that is resistant to various attacks such as replay attacks, man-in-the-middle attacks, impersonation attacks, etc.

B. Security Goals

1) **Confidentiality:** The energy transaction data must not be visible to any unauthorized entity. For this, communication must be secret throughout, i.e., end-to-end. If an unauthorized entity from either within the system such as vehicles authenticated with other aggregators or the currently connected aggregator gains access to the messages which contain energy transaction details, it must be impossible to make sense of it.
2) **Message Integrity:** It must be possible for the smart grid server to verify if the message it receives from the aggregator has been tampered with or compromised. Since EVs and the grid server do not communicate directly, the aggregator must also be able to do the same for the messages received from the EVs.
3) **Identity privacy:** It must be impossible for an unauthorized entity to get hold of any personal information of the vehicle owner of an EV. Even if an unauthorized entity eavesdrops on the data exchanged within the V2G system, it must not be able to figure out that the data is from a particular vehicle or that two transactions are from the same vehicle.
4) **Authentication:** Before any energy transaction can be made, the aggregator must be authenticated with the grid server. The aggregator must also be authenticated with the vehicle, thus preventing any false energy exchanges.

C. Assumptions

The assumptions made in this paper are as follows:

- PUF is a small hardware component that is present with each participating device and is unique.
- The communication between a device and its PUF is secure and tamper-proof.

- The grid is considered as a trusted authority and has sufficient resources. On the other hand, EVs have limited resources in terms of memory and computation power.

VI. Proposed Mutual Authentication Protocol

This section presents the proposed mutual authentication protocol between the vehicle and the grid. Mutual authentication between the vehicle and the grid can be divided as mutual authentication between:

- Aggregator and grid.
- Vehicle and aggregator.

A. Mutual Authentication Between Aggregator and Grid Server

1) When a vehicle wants to make a transaction, the aggregator must authenticate the vehicle. The vehicle sends its ID (ID_V) along with a randomly generated nonce (N_V) to the aggregator with $Msg_{V2M} = \{ID_V, N_V\}$.
2) The aggregator generates another random number, i.e., nonce (N_I), and sends it along with its ID (ID_M) to the grid server with $Msg_{M2G} = \{ID_M, N_I\}$.
3) The first stage of our protocol begins with the aggregator authenticating with the power grid server. This is shown in Figure 2. The grid server receives a message ($Msg_{M2G} = \{ID_M, N_I\}$) from the aggregator. It checks if ID_M exists in its memory and whether N_I is fresh. If either of the conditions fails, the authentication request initiated by aggregator is terminated. Using ID_M, it finds the corresponding set of challenge-response pairs (C, K) (with m pairs) in its memory:

$$C = (C_0, C_1, C_2, \cdots, C_m)$$
$$K = (K_0, K_1, K_2, \cdots, K_m).$$

It also generates a nonce (N_B). To encrypt the message, the server uses a block-based encryption mechanism with m rounds. Let F be any non-linear function which is public to everyone. Thus, even an adversary can know what F is. It can be verified that the security of the protocol does not depend on F. The grid server then computes the following:

$$M_1 = N_I \oplus F(K_0, N_B) \tag{2}$$
$$M_2 = N_B \oplus F(K_1, M_1) \tag{3}$$
$$M_i = M_{i-2} \oplus F(K_{i-1}, M_{i-1}), \;\; 3 \leq i < m \tag{4}$$
$$M_m = M_{m-1} \oplus F(K_{m-1}, M_{m-1}) \tag{5}$$
$$M = (M_{m-1}||M_m) \oplus K_m \tag{6}$$
$$N = m \oplus K_0. \tag{7}$$

4) The grid server sends C, M, N along with a MAC (message authentication code) to the aggregator ID_M, as shown just after the first block under grid server in Figure 2. The MAC is used to verify a few security essentials. The first parameter in the MAC is to identify the correct aggregator. Data integrity is ensured by the second and third parameters. The freshness of the source (grid server in this case) is identified by N_B, which is the last parameter. We use the same approach in the later stages of the protocol as well.
5) On receiving the message from the grid server, aggregator ID_M generates the key K as given in (1) using received challenge C as the input to its PUF. Then, the aggregator calculates m, as shown below:

$$m = N \oplus K_0. \tag{8}$$

6) Using m and K, it then finds N_B as shown in the following equations by applying the same transformations used in the encryption operations in (2)-(5).

$$M_{m-1}||M_m = M \oplus K_m$$
$$M_{i-2} = M_i \oplus F(K_{i-1}, M_{i-1}), \;\; 3 \leq i < m$$
$$N_B = M_2 \oplus F(K_1, M_1)$$
$$N_I = M_1 \oplus F(K_0, N_B).$$

The aggregator uses the MAC to verify the source of the message, checks if its integrity has been compromised, and determines whether the message is fresh or not. If it fails to verify these security traits, authentication is terminated by the aggregator. Else, a nonce N_C is generated. For future authentication, it generates a new set of random challenge-response pairs (C', K') using its PUF:

$$C' = (C'_0, C'_1, C'_2, \cdots, C'_m)$$
$$K' = (K'_0, K'_1, K'_2, \cdots, K'_m).$$

It then calculates M'_i, M'', N' and session key S_k as follows:

$$M'_i = K'_i \oplus K_i$$
$$M'' = M'_0||M'_1||.....||M'_m$$
$$N' = N_C \oplus K_0$$
$$S_k = F(K_0, N_B) \oplus F(K_0, N_C).$$

7) Then, the aggregator sends C', M'', N, N', as well as the MAC to the grid server. Next, it erases interim variables from its memory. This time the MAC includes a fifth parameter which is the session key, S_k. This ensures that both aggregator and grid server have the same session key.
8) On receiving the message from the aggregator, the grid server calculates N_C using N' and K_0:

$$N_C = N' \oplus K_0. \tag{9}$$

Then it calculates K' using M and K:

$$K'_i = M'_i \oplus K_i. \tag{10}$$

The new challenge-response pairs (C', K') are stored in its memory. Then it calculates the session key, S_k, as shown below and verifies the MAC:

$$S_k = F(K_0, N_B) \oplus F(K_0, N_C). \tag{11}$$

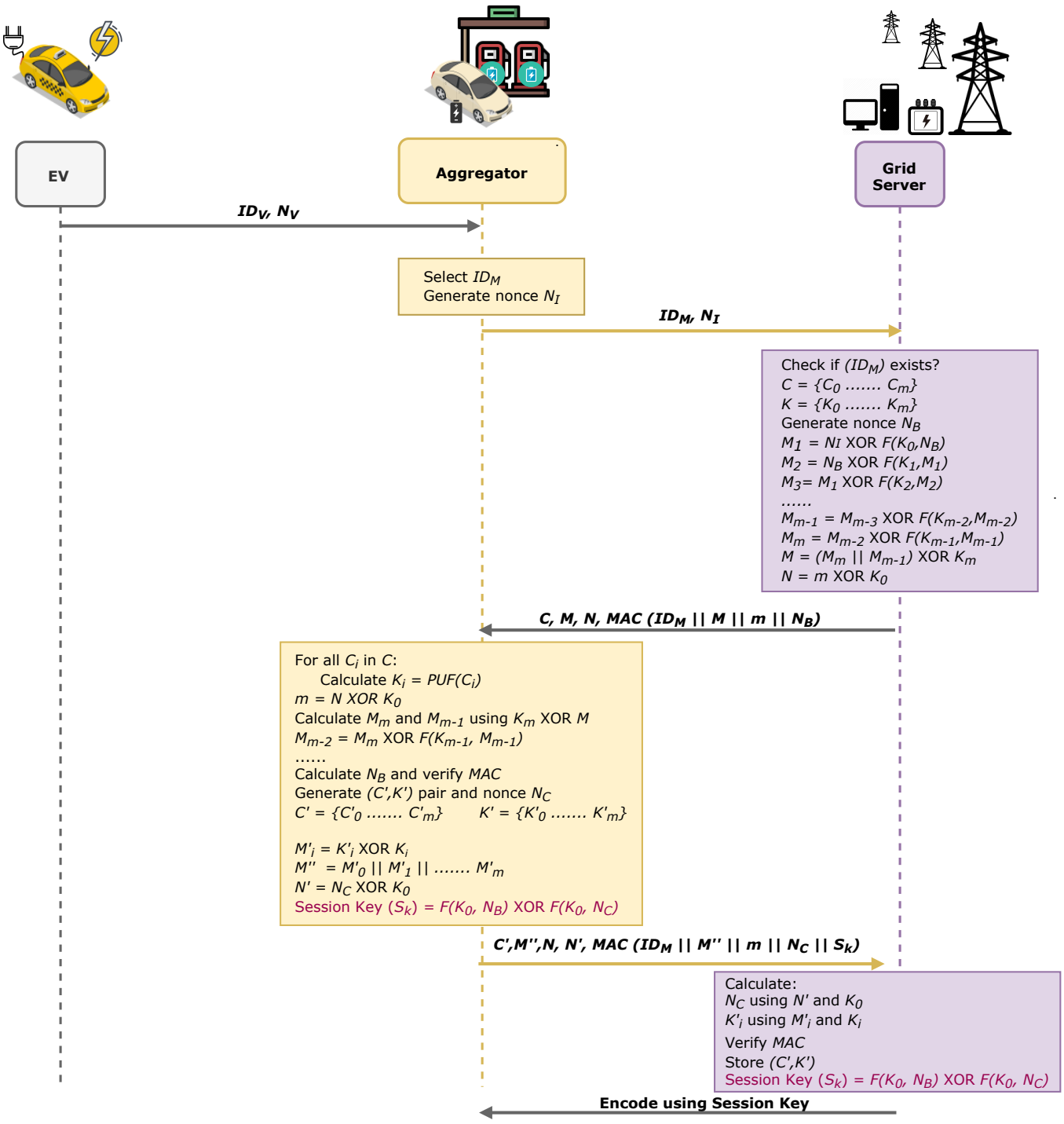


Fig. 2: Mutual authentication between aggregator and the power grid server.

With the session key now established, MA between an aggregator and the grid server is complete.

B. Mutual Authentication between Vehicle and Aggregator

The previous subsection presented the protocol for aggregator and grid server establishing a session key S_k between themselves. This is shown as a small box in Figure 3. Now, we present the protocol for authentication between a vehicle and an aggregator.

1) The aggregator sends an encrypted message $Msg_{M2G} = E([ID_V, N_V], S_k)$ containing the ID of the vehicle ID_V, and its nonce N_V encrypted with S_k to the grid server.
2) The grid server decrypts this message using S_k and obtains ID_V and nonce N_V. It checks within its memory if ID_V exists and whether nonce N_V is fresh. If either of the conditions fails, the authentication request by the vehicle is terminated. Using ID_V, the grid server finds the corresponding set of challenge-response pairs

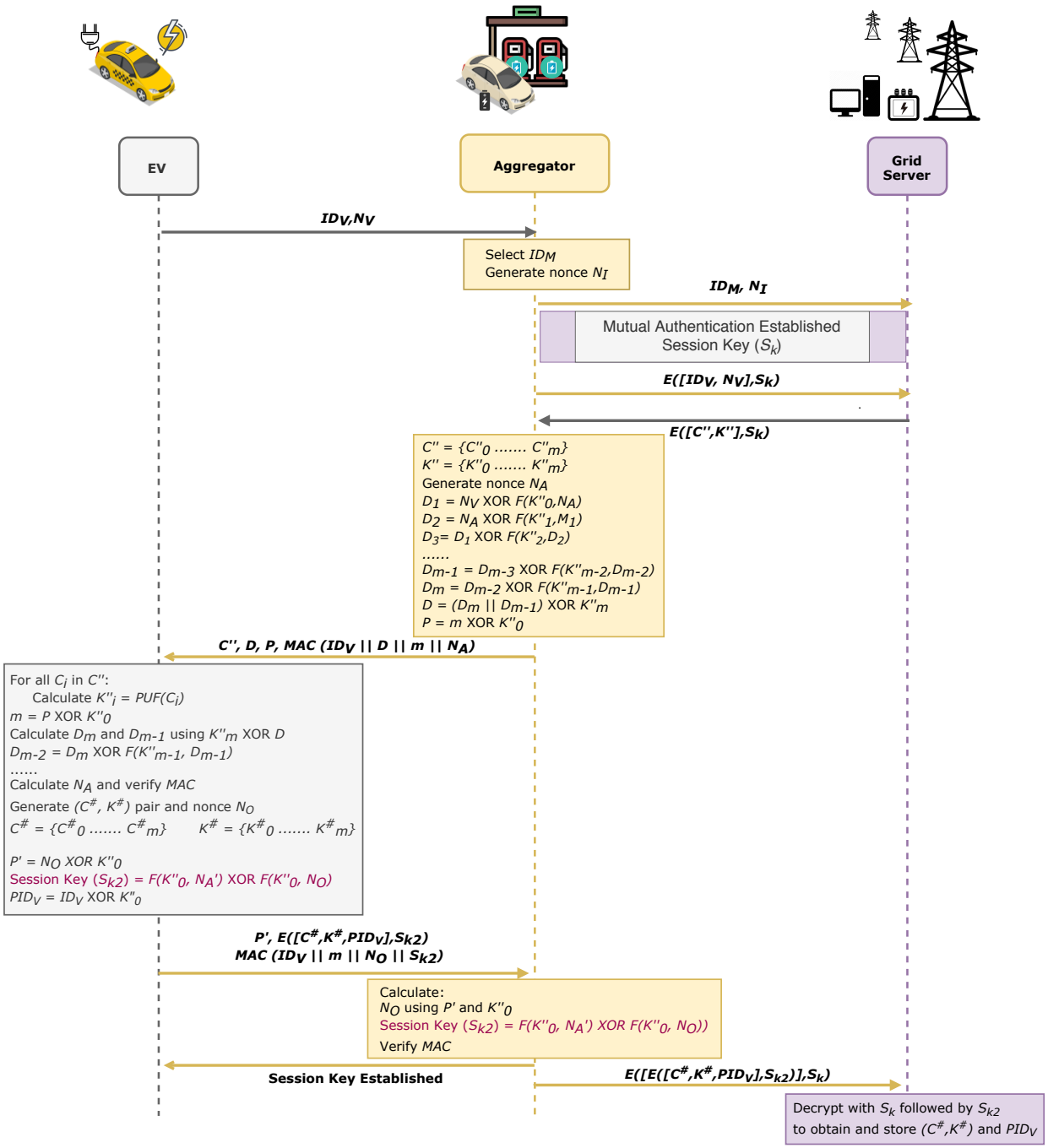


Fig. 3: Mutual authentication between electric vehicle and the aggregator.

(C'', K'') from its memory:

$$C'' = (C''_0, C''_1, C''_2, \cdots, C''_m)$$
$$K'' = (K''_0, K''_1, K''_2, \cdots, K''_m)$$

It then generates a nonce, N_A. Similar to the previous subsection, it uses a block based encryption mechanism to encrypt the message.

$$D_1 = N_V \oplus F(K''_0, N_A)$$
$$D_2 = N_A \oplus F(K''_1, D_1)$$
$$D_i = D_{i-2} \oplus F(K''_{i-1}, D_{i-1}), \ \ 3 \leq i < m$$
$$D_m = D_{m-2} \oplus F(K''_{m-1}, D_{m-1})$$
$$D = (D_m || D_{m-1}) \oplus K''_m)$$
$$P = m \oplus K''_0.$$

3) The aggregator sends C'', D, P and the MAC to the EV. Within the MAC, the first parameter verifies the identity of the vehicle. Data integrity is ensured by the second and third parameters. Freshness of the source (aggregator in

this case) is identified by N_A which is the last parameter.

4) On receiving the message from the aggregator, the vehicle generates the key K'' by using its PUF for the newly received challenge C'' as given in (1). Then, it calculates m as shown below:

$$m = P \oplus K''_0. \tag{12}$$

Using m and K, it finds N_A as shown:

$$D_m||D_{m-1} = K''_m \oplus D$$
$$D_{i-2} = D_i \oplus F(K''_{i-1}, D_{i-1})$$
$$N_A = D_2 \oplus F(K''_1, M_1)$$
$$N_V = D_1 \oplus F(K''_0, N_A).$$

5) The vehicle uses the MAC to verify the source of the message, checks if its integrity has been compromised, and determines whether the message is fresh or not. If it fails to verify these security traits, authentication is terminated by the vehicle. Else, a nonce N_O is generated by the vehicle. For future authentication it generates a new set of challenge-response pairs using its PUF:

$$C^{\#} = (C^{\#}_0, C^{\#}_1, C^{\#}_2, \cdots, C^{\#}_m)$$
$$K^{\#} = (K^{\#}_0, K^{\#}_1, K^{\#}_2, \cdots, K^{\#}_m).$$

It then calculates P' and session key S_{k2} as follows:

$$P' = N_O \oplus K''_0$$
$$S_{k2} = F(K''_0, N_A) \oplus F(K''_0, N_O)).$$

The EV then calculates its new pseudonym or pseudo-ID PID_V to be used the next time it wants to authenticate:

$$PID_V = ID_V \oplus K''_0. \tag{13}$$

This ensures identity protection because an adversary will not be able to figure out whether a previous transaction belonged to the same EV or not. The EV sends P', $E([C^{\#}, K^{\#}, PID_V], S_{k2})$ and a MAC to the aggregator ID_M. This time the session key S_{k2} is used as a parameter in the MAC to ensure that both EV and aggregator have the same session key.

6) On receiving the message from the vehicle, the aggregator obtains N_O using P and K''_0:

$$N_O = P' \oplus K''_0. \tag{14}$$

Then, it calculates the session key S_{k2} as shown below and verifies the MAC:

$$S_{k2} = F(K''_0, N_A) \oplus F(K''_0, N_O). \tag{15}$$

The newly generated challenge-response pair of the PUF in the EV and the pseudo-ID of the EV are sent to the grid server in the following message Msg_{M2G}.

$$Msg_{M2G} = E([E([C^{\#}, K^{\#}, PID_V], S_{k2}), S_{k2}], S_k) \tag{16}$$

7) The grid server decrypts the message with S_k to obtain $E([C^{\#}, K^{\#}, PID_V], S_{k2})$ and S_{k2}. Next, it decrypts $E([C^{\#}, K^{\#}, PID_V], S_{k2})$ with S_{k2} to obtain and store in its memory the new challenge-response pair of the vehicle $(C^{\#}, K^{\#})$, and the new pseudo-ID PID_V. If any hijacker tries to tamper with the aggregator device, its PUF will be destroyed and the protocol will not proceed to this stage. Therefore, the adversary will not be able to access the new pseudo-ID.

VII. Security Analysis

In this section, we formally show that our MA protocol is secure. We use Mao and Boyd logic [42] which is extensively used for security analysis of protocols. In our analysis, we denote vehicle ID_V, aggregator ID_M, and the grid server by V, M, and G, respectively.

A. Mao-Boyd Logic

The basic building blocks of Mao-Boyd Logic listed below are necessary to understand the protocol verification.

1) $A \models B$: A believes B is legitimate and that it may function correspondingly.
2) $A \overset{K}{\sim} B$: A encrypted B using key K.
3) $A \overset{K}{\triangleleft} B$: A sees B using decipherment key K.
4) $A \overset{K}{\leftrightarrow} B$: K is a valid shared key between entities A and B.
5) $\#(N)$: Nonce N is new and fresh.
6) $sup(P)$: P is a credible and reliable entity.
7) $A \triangleleft \parallel M$: Entity A does not have access to message M.

In our proof we use several inference rules of Mao and Boyd logic which are listed in Table II. In the rules, '$\bigwedge$' represents the logical AND of two statements. If P,Q are statements, and the inference of their logical AND is statement R, it is written in Mao and Boyd logic as shown below.

$$\frac{P \bigwedge Q}{R}$$

B. New Security Analysis for protocol

First, let us consider the MA between an aggregator and the power grid server. We first prove the statement "M is convinced N_B is a valid shared key between M and G". The following proof is summarized in Mao and Boyd logic in Fig. 5a. The challenge-response pair of M, (C, K) is stored in G, therefore it can be said that "M and G have a well-kept secret K". In Mao and Boyd logic, this is written as shown in equation (i). Using K, the aggregator is able to decipher the variable M in message 3 of the protocol and obtain N_B and N_I. Therefore, "M sees N_I with decipherment key K" which is (ii) and "M sees N_B with decipherment key K" which is (vi).

$$M \models M \overset{K}{\leftrightarrow} G \tag{i}$$
$$M \overset{K}{\triangleleft} N_I \tag{ii}$$

Applying the ***authentication rule*** to statements (i) and (ii), we obtain "M believes G encrypted N_I using key K" which

TABLE II

Name	Inference Rule
Authentication rule	$\frac{P\models P\overset{K}{\leftrightarrow}Q \wedge P\overset{K}{\triangleleft}M}{P\models Q\overset{K}{\sim}M}$
Nonce-verification rule	$\frac{P\models\#(M) \wedge P\models Q\overset{K}{\sim}M}{P\models Q\models P\overset{K}{\leftrightarrow}Q}$
Confidentiality rule	$\frac{P\models P\overset{K}{\leftrightarrow}Q \wedge P\models S^c\triangleleft\|M \wedge P\overset{K}{\sim}M}{P\models(S\cup\{Q\})^c\triangleleft\|M}$
Super-principal rule	$\frac{P\models Q\models X \wedge P\models sup(Q)}{P\models X}$
Intuitive rule	$\frac{P\overset{K}{\triangleleft}M}{P\triangleleft M}$
Good Key rule	$\frac{P\models\{P,Q\}^c\triangleleft\|K \wedge P\models\#(K)}{P\models P\overset{K}{\leftrightarrow}Q}$
Fresh rule	$\frac{P\models\#(M) \wedge P\triangleleft N\mathbf{R}M}{P\models\#(N)}$

is (iii). Since M generates a new nonce N_I each time, we can say "M believes N_I is new and fresh" which is (iv). On applying the ***nonce-verification rule*** to (iii) and (iv) we obtain (v) which is "M is convinced that G is convinced that K is a well-kept secret between M and G".

$$M \models G \overset{K}{\sim} N_I \quad \text{(iii)}$$

$$M \models \#(N_I) \quad \text{(iv)}$$

$$M \models G \models M \overset{K}{\leftrightarrow} G \quad \text{(v)}$$

We then apply ***authentication rule*** to (i) and (vi), to obtain (vii) which is "M is convinced that G encrypted N_B using K". Since G generates a new nonce N_B each time, M is knows that no one apart from G could have seen N_B. Thus, we have the statement "M is convinced that G is convinced that no one other than M has access to N_B" which is (viii). Applying the ***confidentiality rule*** to (v), (vii) and (viii) we get (ix) which states "M is convinced that G is convinced that no one other than M and B has access to N_B".

$$M \overset{K}{\triangleleft} N_B \quad \text{(vi)}$$

$$M \models G \overset{K}{\sim} N_B \quad \text{(vii)}$$

$$M \models G \models \{M\}^c\triangleleft \parallel N_B \quad \text{(viii)}$$

$$M \models G \models \{M, G\}^c\triangleleft \parallel N_B \quad \text{(ix)}$$

It is assumed in the protocol that G is a credible and reliable entity and M believes this as fact. Hence, the statement "M believes that G is a credible and reliable entity (super-principal)" which is (x). Next, we apply the ***super-principal rule*** to statements (ix) and (x), to obtain (xi) which is "M is convinced that no one other than M and G has access to N_B".

To proceed further we need to understand a few definitions and rules of message idealization from [42] which are discussed in the Appendix.

In message 2 of the Fig. 2 M sends N_I to G. As a response, G sends N_B in message 3 by encrypting it inside variable M. By deciphering variable M, G obtains nonces N_I and N_B. Therefore, according to the message idealization rules presented in the appendix, N_I can be considered as a challenge and N_B can be considered its response. Note that these are not the same challenge-response pair (C, K) of the PUF. Thus we arrive at the statement "M can see the replied challenge N_I and the response N_B with decipherment key K" which is (xii). On applying the ***intuitive rule*** to (xii), we get (xiii) which is "M can see the replied challenge N_I and the response N_B".

$$M \models sup(G) \quad \text{(x)}$$

$$M \models \{M, G\}^c\triangleleft \parallel N_B \quad \text{(xi)}$$

$$M \overset{K}{\triangleleft} N_I \ \mathbf{R}\ N_B \quad \text{(xii)}$$

$$M \triangleleft N_I \ \mathbf{R}\ N_B \quad \text{(xiii)}$$

We then apply the ***fresh rule*** to (iv) and (xiii), we obtain statement (xiv) which is "M believes N_B is new and fresh".

$$M \models \#(N_B) \quad \text{(xiv)}$$

$$M \models M \overset{N_B}{\leftrightarrow} G \quad \text{(xv)}$$

Finally, we applying the ***good-key rule*** to statements (x), (xi) and (xiv) to prove the statement "M is convinced that N_B is a valid shared key between M and G".

In a similar manner the proof for "G is convinced that N_B is a valid shared key M and G" is shown in Fig. 5d. The statements "M is convinced that N_C is a valid shared key between M and G" and "G is convinced that N_C is a valid shared key between M and G" are shown in Fig. 5b and Fig. 5c respectively. The statements "M is convinced that K' is a valid secret key between M and G" and "G is convinced that K' is a valid secret key between M and G" are shown in Fig. 5f and Fig. 5e respectively. In these figures, the logical AND operation between two statements is represented

by a '$\bigwedge$'. Thus, we have shown that an adversary cannot see N_B, N_C or K'. The three variables N_B, N_C and K' are critical because without them an adversary cannot decipher the communicated data. The Mao Boyd formal proof discussed above has proven the secrecy of N_B, N_C and K' which is regardless of the kind of attack used by the adversary such as man-in-the-middle (MITM) attack, masquerade attack, or replay attack. Note that even if an attacker physically hijacks the aggregator or the EV, by virtue of the property of PUF discussed in section I, it is ensured that the adversary cannot obtain the legitimate challenge-response pairs. Additionally, there are no secrets stored on the aggregator or the EV itself. Thus, physical security and protection against node tampering attack are also guaranteed. Untraceablity is ensured by using a pseudo-ID as in equation 13 guarantees untraceability.

The parameters used in generating the session key S_k are K_0, N_B and N_C and the parameters used in generating session key S_{k2} are K_0'', N_A and N_O. The nonces N_B, N_C, N_A and N_O are cryptographic nonces which are randomly generated in every session. As already discussed in section VI, the freshness of the nonce is a necessary condition which is checked at several stages of the MA protocol. Unless freshness is verified, MA does not take place. In addition, the challenge response pairs used in the protocol (C, K), (C', K'), $(C", K")$ and $(C^{\#}, K^{\#})$ are all randomly generated. First a challenge is randomly generated and its corresponding PUF response is obtained. The output of a PUF depends both on the physical disorder as well as the applied challenge, hence the response obtained for each challenge will not only be random, but also very different from each other. Therefore K_0 and K_0'' will change randomly in each authentication round. The combined effect of the randomness of these variables and the non-linearity of the function F guarantee that a unique session key is obtained in each round for both stages of the protocol.

Similarly, the Mao and Boyd logic proofs for the MA between the EV and the aggregator are shown in Fig. 5. In this case it is established that the critical variables of this stage, i.e, N_A, N_O and K'' cannot be obtained by an adversary.

VIII. Comparison and Analysis

A. Security Goals And Protection Against Various Attacks

A comparison of the security features of our protocol with a different state of the art protocols currently in use in V2G systems is presented in Table III. "✓" indicates that the protocol possesses a feature or is secure against an attack. A blank indicates that the protocol lacks a feature or is insecure against an attack. All the mentioned protocols provide MA except [24]. Without MA, a participating entity can neither verify if it is sending a message to a trusted entity, nor can it verify if the message it received is from a trusted entity. With MA, both the sending and receiving parties can be sure of each other's authenticity. Identity protection is not provided by the protocol in [23]. Consequently, an attacker may easily figure out the real identity of the EV by looking at the usage data. The protocols in [20] and [22] do not provide message integrity. Our protocol uses MAC to ensure this. All the entities (EVs, aggregators and grid server) can easily detect any tampering in the messages they receive. The protocol in [20] is vulnerable to man-in-the-middle attacks. An adversary may insert itself between the communication of an EV and the aggregator, or between the aggregator and the grid server and gain control of the communication between them. The protocols in [19], [20] and [22] are vulnerable against impersonation attacks. The protocols in [20] and [22] are not secure against replay attacks. The protocols in [20] and [23] do not provide session key security. Physical security is provided only by the proposed protocol (SUKA). As mentioned in Section V-B, an attacker that captures an EV device must not be able to gather any secrets. As also mentioned in Section I, almost all authentication protocols proposed in the literature require that the EVs store at least one secret in their memory, if not more. Such storing of secrets on any device renders the protocols vulnerable to physical attacks. The MA protocol proposed in this paper has two features which make it resistant to any physical attacks: (i) EVs and aggregators need not store any secrets in their memory; (ii) there is a secure communication between the EV's microcontroller and its PUF since they are both on the same chip [43]. Thus, even though an attacker may physically capture the device, it would be impossible for them to extract any secrets. Therefore, SUKA is resilient against physical attacks. The papers in [19], [20] [22], [24] and [31] do not provide a formal security proof for their proposed protocols.

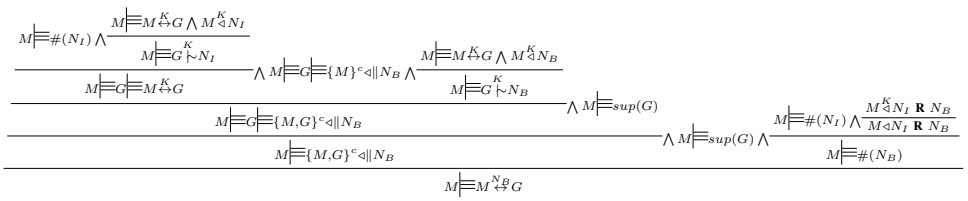
B. Computation Overhead

In Table IV, we present a comparison of the computation costs of our protocol with some state-of-the-art protocols which have a similar system model as ours. We show the comparison for the case where one EV is authenticating with the grid.

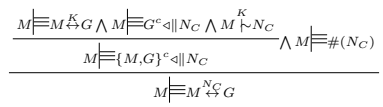
In Table IV, the number of cryptographic operations, pairing operations, encryption/decryption, hash operations, MAC computations and PUF executions are listed for one round of authentication. Our protocol uses only 33 cryptographic operations (which include XOR, addition, scalar multiplication and exponential computation) compared to 37 in [30] and 36 in [20]. Our protocol uses zero pairing operations. While [20] has only 2 encryption/decryption operations and 4 MAC/HMAC computations, it has 9 hash function computations while ours has zero. Although [30] has no encryption/decryption operations or MAC/HMAC computations, it has 16 hash computations while ours has none. While there is no physical security in [19], [20] and [30], our protocol is physically secured by the use of PUFs, which requires 2 operations. We argue that the overall performance of our protocol is much better due to lesser computation overhead and far superior security features.

C. Performance Comparison

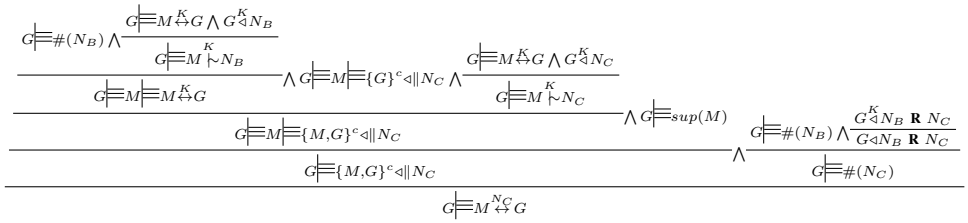
We simulated the operations carried out by an EV in the security schemes of [19], [30] and [20], all of which have a similar system model as SUKA. The simulations were carried out in Python 2.7 on a PC with Intel Core i5-5200U processor



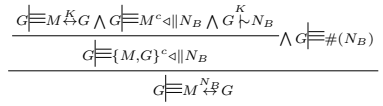
(a) Proof for: "**M** is convinced that N_B as a valid shared key between **M** and **G**".



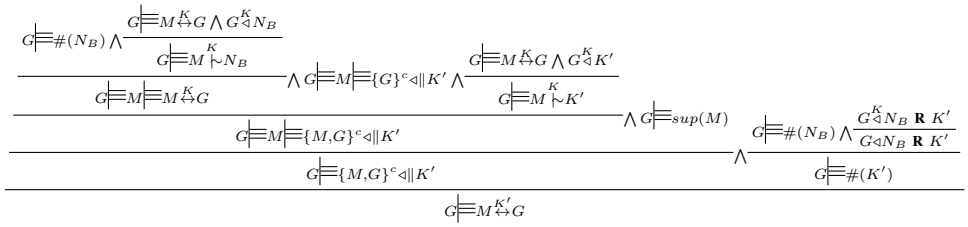
(b) Proof for: "**M** is convinced that N_C is a valid shared key between **M** and **G**".



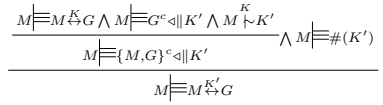
(c) Proof for: "**G** is convinced that N_C is a valid shared key between **M** and **G**".



(d) Proof for: "**G** is convinced that N_B is a valid shared key between **M** and **G**".

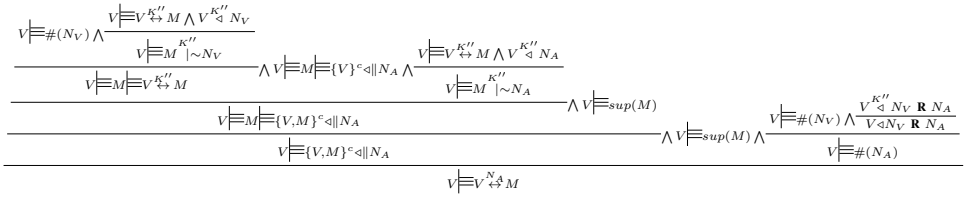


(e) Proof for: "**G** is convinced that K' is a valid shared key between **M** and **G**".

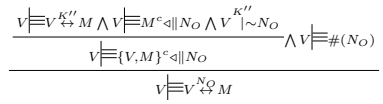


(f) Proof for: "**M** is convinced that K' is a valid shared key between **M** and **G**".

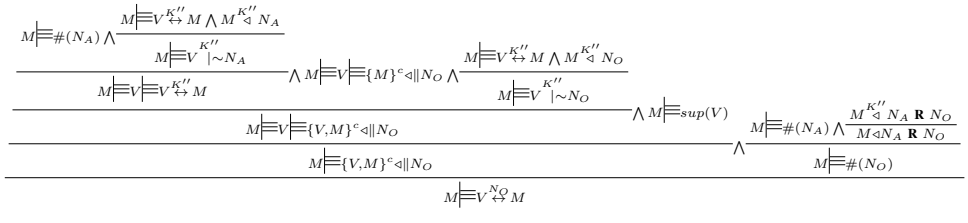
Fig. 4: Proof for authentication between aggregator and power grid server



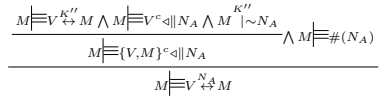
(a) Proof for: "**V** is convinced that N_A as a valid shared key between **V** and **M**".



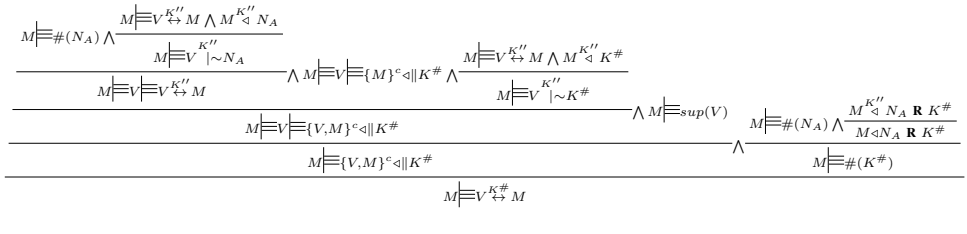
(b) Proof for: "**V** is convinced that N_O is a valid shared key between **V** and **M**".



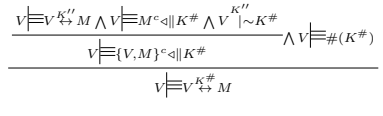
(c) Proof for: "**M** is convinced that N_O is a valid shared key between **V** and **M**".



(d) Proof for: "**M** is convinced that N_A is a valid shared key between **V** and **M**".



(e) Proof for: "**M** is convinced that $K^{\#}$ is a valid shared key between **V** and **M**".



(f) Proof for: "**V** is convinced that $K^{\#}$ is a valid shared key between **V** and **M**".

Fig. 5: Proof for authentication between EV and aggregator

TABLE III: Comparison of Security Features

Features	[18]	[19]	[20]	[22]	[23]	[24]	[31]	SUKA
Mutual Authentication	✓	✓	✓	✓	✓		✓	✓
Identity Protection	✓	✓	✓	✓		✓	✓	✓
Message Integrity	✓	✓			✓	✓	✓	✓
Man-In-The-Middle Attack	✓	✓		✓	✓	✓	✓	✓
Impersonation Attack	✓				✓	✓	✓	✓
Replay Attack	✓	✓			✓	✓	✓	✓
Session Key Security	✓	✓		✓		✓	✓	✓
Physical Security								✓
Formal Security Proof	✓				✓			✓

TABLE IV: Comparison of computation overhead

Operations	[19]	[20]	[30]	SUKA
Cryptographic operations ($\oplus, +$, scalar multiplication and exponent)	81	36	37	33
Pairing	19	-	-	-
Encryption/Decryption	-	2	-	6
Hash (H)	6	9	16	-
MAC/HMAC	7	4	-	8
PUF	-	-	-	2

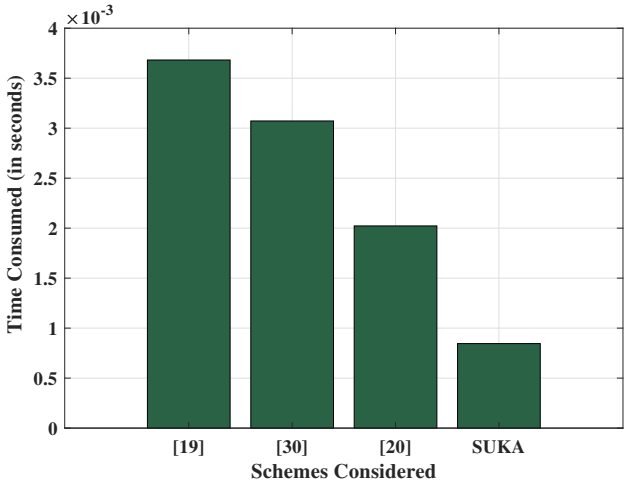


Fig. 6: Comparison of time consumed by EV for MA in SUKA and the security schemes of [19], [30] and [20].

with 8GB DDR3 RAM. Fig. 6, shows the time consumed by the EV in every round of authentication in the considered schemes. The EV consumes 3.682 ms, 3.072 ms, 2.022 ms in the security schemes of [19], [30] and [20] respectively whereas only 0.845 ms in SUKA. Therefore, SUKA is more efficient than state-of-the-art security schemes.

IX. Conclusion

This paper proposed MA protocols for the two stages or steps which arise in a V2G system: (i) For MA between the aggregator and the grid server, and (ii) for MA between EV and aggregator. The proposed protocol (SUKA) uses a challenge-response architecture, which is enabled by PUFs. This gives our proposed protocol the advantage of not having to store any secret information in EVs and aggregators. Secrets are stored only in the grid server. Only one set of challenge-response pairs is stored in the server for every EV. Two session keys are established when an EV wants to authenticate with the grid server: one session key between the aggregator and the grid server, and another one between the EV and the aggregator. We showed that SUKA achieves MA, identity protection, message integrity, physical security, and session key security along with protection against various attacks such as MITM attacks, replay attacks and impersonation attacks. SUKA is proven formally secure by Mao and Boyd logic and uses simple computations, which makes it very efficient and fast. Hence, the proposed protocol is a viable solution for upcoming V2G systems.

Appendix

A. Rules of Message Idealization

- A message without any symbols is called an *atomic message (AM)*.
- If an *AM* is sent at one stage of the protocol by a node and received by the same node in another stage of the protocol, it is called a *challenge*.
- A *challenge* sent to its originating node is called a *replied challenge*.
- If an *AM* and a *response* are sent together by a single node for the first time, it is called a *response*.
- *AMs* which are not *challenges* or *responses* are treated as nonsense and are discarded.
- In case an *AM* qualifies as a *challenge* and *response* in a single line, it is considered a *response*.
- A *replied challenge* and its *response* together is denoted as *Response* ***R*** *RC*.

References

[1] B. K. Sovacool and R. F. Hirsh, "Beyond batteries: An examination of the benefits and barriers to plug-in hybrid electric vehicles (phevs) and a vehicle-to-grid (v2g) transition," *Energy Policy*, vol. 37, no. 3, pp. 1095–1103, 2009.

[2] C. D. White and K. M. Zhang, "Using vehicle-to-grid technology for frequency regulation and peak-load reduction," *Journal of Power Sources*, vol. 196, no. 8, pp. 3972–3980, 2011.

[3] H. Liu, Z. Hu, Y. Song, and J. Lin, "Decentralized vehicle-to-grid control for primary frequency regulation considering charging demands," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3480–3489, 2013.

[4] H. Lund and W. Kempton, "Integration of renewable energy into the transport and electricity sectors through v2g," *Energy policy*, vol. 36, no. 9, pp. 3578–3587, 2008.

[5] L. Gelazanskas and K. A. Gamage, "Demand side management in smart grid: A review and proposals for future direction," *Sustainable Cities and Society*, vol. 11, pp. 22–30, 2014.

[6] H. Wang, B. Qin, Q. Wu, L. Xu, and J. Domingo-Ferrer, "Tpp: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2340–2351, 2015.

[7] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[8] G. K. Verma, B. B. Singh, N. Kumar, and V. Chamola, "CB-CAS: Certificate-Based Efficient Signature Scheme with Compact Aggregation for Industrial Internet of Things Environment," *IEEE Internet of Things Journal*, vol. PP, no. c, pp. 1–1, 2019.

[9] A. Abdallah and X. Shen, "Lightweight Security and Privacy-Preserving Scheme for V2G Connection," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, dec 2015, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7417592/

[10] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications," in *2008 IEEE International Conference on RFID*. IEEE, apr 2008, pp. 58–64. [Online]. Available: https://ieeexplore.ieee.org/document/4519377/

[11] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Brand and IP protection with physical unclonable functions," in *2008 IEEE International Symposium on Circuits and Systems*. IEEE, may 2008, pp. 3186–3189. [Online]. Available: http://ieeexplore.ieee.org/document/4542135/

[12] W. Kempton and J. T. Tomic, "Vehicle-to-grid power fundamentals: Calculating capacity and net revenue," *Journal of Power Sources*, vol. 144, pp. 268–279, 2005. [Online]. Available: http://www.udel.edu/V2G.

[13] Sekyung Han, Soohee Han, and K. Sezaki, "Development of an Optimal Vehicle-to-Grid Aggregator for Frequency Regulation," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 65–72, jun 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5446440/

[14] F. Kennel, D. Gorges, and S. Liu, "Energy management for smart grids with electric vehicles based on hierarchical MPC," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1528–1537, 2013.

[15] C. Guille and G. Gross, "A conceptual framework for the vehicle-to-grid (V2G) implementation," *Energy Policy*, vol. 37, no. 11, pp. 4379–4390, nov 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421509003978

[16] B. K. Sovacool and R. F. Hirsh, "Beyond batteries: An examination of the benefits and barriers to plug-in hybrid electric vehicles (PHEVs) and a vehicle-to-grid (V2G) transition," 2008. [Online]. Available: www.elsevier.com/locate/enpol

[17] L. Pieltain Fernández, T. Gómez San Román, R. Cossent, C. Mateo Domingo, and P. Frías, "Assessment of the impact of plug-in electric vehicles on distribution networks," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 206–213, 2011.

[18] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos, "Network Security and Privacy Challenges in Smart Vehicle-to-Grid," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 88–98, 2017.

[19] Z. Yang, S. Yu, W. Lou, and C. Liu, "P^{2}: Privacy-Preserving Communication and Precise Reward Architecture for V2G Networks in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 697–706, dec 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5771586/

[20] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, "Aggregated-proofs based privacy-preserving authentication for V2G networks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1722–1733, 2012.

[21] H.-R. Tseng, "A secure and privacy-preserving communication protocol for V2G networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, apr 2012, pp. 2706–2711. [Online]. Available: http://ieeexplore.ieee.org/document/6214259/

[22] H. Liu, H. Ning, Y. Zhang, Q. Xiong, and L. T. Yang, "Role-dependent privacy preservation for secure v2g networks in the smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 208–220, feb 2014.

[23] J. L. Tsai and N. W. Lo, "Secure Anonymous Key Distribution Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 906–914, mar 2016.

[24] A. Abdallah and X. Shen, "Lightweight Authentication and Privacy-Preserving Scheme for V2G Connections," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2615–2629, 2017.

[25] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably Secure Authenticated Key Agreement Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1900–1910, may 2018.

[26] J. Shen, T. Zhou, F. Wei, X. Sun, and Y. Xiang, "Privacy-preserving and lightweight key agreement protocol for V2G in the social internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2526–2536, aug 2018.

[27] H. Guo, Y. Wu, F. Bao, H. Chen, and M. Ma, "UBAPV2G: A unique batch authentication protocol for vehicle-to-grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 707–714, 2011.

[28] H. Liu, H. Ning, Y. Zhang, and M. Guizani, "Battery status-aware authentication scheme for V2G networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 99–110, 2013.

[29] J. Chen, Y. Zhang, and W. Su, "An anonymous authentication scheme for plug-in electric vehicles joining to charging/discharging station in vehicle-to-Grid (V2G) networks," *China Communications*, vol. 12, no. 3, pp. 9–19, 2015.

[30] N. Saxena and B. J. Choi, "Authentication Scheme for Flexible Charging and Discharging of Mobile Vehicles in the V2G Networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1438–1452, jul 2016.

[31] M. Tao, K. Ota, M. Dong, and Z. Qian, "AccessAuth: Capacity-aware security access authentication in federated-IoT-enabled V2G networks," *J. Parallel Distrib. Comput.*, vol. 118, pp. 107–117, 2018. [Online]. Available: https://doi.org/10.1016/j.jpdc.2017.09.004

[32] P. Gope and B. Sikdar, "Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart Grids," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1554–1566, jun 2019.

[33] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[34] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The Bistable Ring PUF: A new architecture for strong Physical Unclonable Functions," in *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, jun 2011, pp. 134–141. [Online]. Available: http://ieeexplore.ieee.org/document/5955011/

[35] T. Alladi, V. Chamola, B. Sikdar, and K.-k. R. Choo, "Consumer IoT : Security Vulnerability Case Studies and Solutions Consumer IoT : Security Vulnerability Case Studies and Solutions," no. October, 2019.

[36] S. Pandey, S. Deyati, A. Singh, and A. Chatterjee, "Noise-resilient sram physically unclonable function design for security," in *2016 IEEE 25th Asian Test Symposium (ATS)*. IEEE, 2016, pp. 55–60.

[37] D. Jeon, J. H. Baek, D. K. Kim, and B.-D. Choi, "Towards zero bit-error-rate physical unclonable function: Mismatch-based vs. physical-based approaches in standard cmos technology," in *2015 Euromicro Conference on Digital System Design*. IEEE, 2015, pp. 407–414.

[38] K.-H. Chuang, E. Bury, R. Degraeve, B. Kaczer, D. Linten, and I. Verbauwhede, "A physically unclonable function using soft oxide breakdown featuring 0% native ber and 51.8 fj/bit in 40-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, 2019.

[39] X. Lu, L. Hong, and K. Sengupta, "Cmos optical pufs using noise-immune process-sensitive photonic crystals incorporating passive variations for robustness," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2709–2721, 2018.

[40] W.-C. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, "Design and analysis of stability-guaranteed pufs," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 978–992, 2017.

[41] D. MRaihi, S. Machani, M. Pei, and J. Rydell, "Rfc 6238-totp: Time-based one-time password algorithm," *Internet Requests for Comments*, 2011.

[42] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *[1993] Proceedings Computer Security Foundations Workshop VI*. IEEE Comput. Soc. Press, pp. 147–158. [Online]. Available: http://ieeexplore.ieee.org/document/246631/

[43] S. Sutar, A. Raha, and V. Raghunathan, "Memory-based combination pufs for device authentication in embedded systems," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 793–810, 2018.