Dustin Kern*, Timm Lauser, and Christoph Krauß

Integrating Privacy into the Electric Vehicle Charging Architecture

Abstract: The Electric Vehicle (EV) charging architecture consists of several actors which communicate with different protocols. A serious issue is the lack of adequate privacy-preserving measures that enables the generation of movement profiles or inferring consumer habits by all of the involved actors. In this paper, we propose an extension of a Trusted Platform Module (TPM)-based Direct Anonymous Attestation (DAA) scheme to enable privacy-preserving charging authorization and billing. Our implementation shows that our solution can be easily integrated into existing protocols of the Plug-and-Charge (PnC) EV charging architecture and introduces only minor overhead. The formal analysis using the Tamarin prover shows the security and privacy of our protocol extension.

Keywords: Electric Vehicle, Privacy, Direct Anonymous Attestation, Trusted Platform Module

DOI 10.56553/popets-2022-0066
Received 2021-11-30; revised 2022-03-15; accepted 2022-03-16.

Acronyms

CCH	Contract Clearing House
CDR	Charge Detail Record
CP	Charge Point
CPO	Charge Point Operator
CPS	Certificate Provisioning Service
DAA	Direct Anonymous Attestation
ECC	Elliptic Curve Cryptography
eMAID	e-Mobility Account Identifier
eMSP	e-Mobility Service Provider
EV	Electric Vehicle
HMAC	Hash-based Message Authentication Code
OEM	Original Equipment Manufacturer
OCPI	Open Charge Point Interface
OCPP	Open Charge Point Protocol
PCID	Provisioning Certificate Identifier
PnC	Plug-and-Charge
SIA	Subject Information Access
TLS	Transport Layer Security
TPM	Trusted Platform Module

***Corresponding Author: Dustin Kern:** Darmstadt University of Applied Sciences, E-mail: dustin.kern@h-da.de
Timm Lauser: Darmstadt University of Applied Sciences, E-mail: timm.lauser@h-da.de
Christoph Krauß: Darmstadt University of Applied Sciences, E-mail: christoph.krauss@h-da.de

1 Introduction

The adoption of Electric Vehicles (EVs) is steadily increasing worldwide and EVs are not only charged at home but often also at public Charge Points (CPs). Today, authorization and billing are mostly realized via the EV user's RFID card that is presented at the CP. A more user-friendly alternative to RFID cards is provided by Plug-and-Charge (PnC) based on ISO 15118 [27, 28], in which an EV uses locally installed credentials to automatically authenticate itself at a CP for charge authorization and billing.

The EV charging architecture involves a multitude of actors. For charge billing, the EV user has a contract with an e-Mobility Service Provider (eMSP) and is provided with the means for charge authorization (e.g., an RFID card or PnC credentials). The EV of the user is charged at a CP which is operated by a Charge Point Operator (CPO). In case of roaming, i.e., the EV is charged at a CP for which its contract does not directly apply, a Contract Clearing House (CCH) acts as intermediary between a multitude of CPOs and eMSPs.

Communication between these actors involves a variety of personal data, e.g., unique identifiers, amount of charged energy, or time and location of the charging session. However, existing communication protocols currently in use do not support any privacy measures and actors receive more data than required for performing their tasks. This allows an adversary to create movement profiles or infer consumer habits. To do this, an attacker can, for example, attack communications or compromise systems such as CPs. That such attacks are realistic was already shown in 2017 in [8]. Also, the operators of the various backend systems can be attackers, since they may be interested in analyzing data and us-

ing it for their own purposes. Thus, privacy-preserving measures are required to also be compliant with the European General Data Protection Regulation.

In this paper, we propose an approach for privacy-preserving charging authorization and billing that can extend existing protocols of the PnC EV charging architecture. Our approach guarantees the anonymity of the EV (and its driver) whenever possible and ensures unlinkability of charging sessions. The contributions of this paper are as follows: (i) Identification of the relevant personal data and the actors that may access this data if necessary for their operation. (ii) Derivation of security and privacy requirements using STRIDE [22] and LINDDUN [10] analyses as well as functional requirements. (iii) Extension of the Trusted Platform Module (TPM)-based Direct Anonymous Attestation (DAA) scheme proposed in [61] to enable the integration into existing PnC protocols. (iv) Proof-of-concept implementation showing minor additional overhead and easy integration into existing systems with minimal protocol changes. (v) Formal security and privacy analysis of our extended Direct Anonymous Attestation (DAA) scheme in the symbolic model using the Tamarin prover [40].

The remainder of this paper is structured as follows: In Section 2, we introduce our considered system model and in Section 3 the identified requirements. We describe our concept for a privacy-preserving PnC protocol extension in Section 4 and the proof-of-concept implementation with functional evaluation in Section 5. The formal verification with regard to the security and privacy requirements is presented in Section 6. In Section 7, we distinguish our work from related work. Finally, we conclude the paper in Section 8.

2 System Model

A high-level view of the contract-based EV charging system model is provided in Fig. 1, including the actors as outlined in Section 1. The communication between EV and CP uses the ISO 15118 protocol and the CP communicates with its CPO via Open Charge Point Protocol (OCPP) [43] as the de facto standard [13]. Several competing protocols exist for the roaming communication between CPOs, CCHs, and eMSPs [13]. We use Open Charge Point Interface (OCPI) as exemplary roaming protocol since [58] ranks it the highest in terms of transparency, openness, and impartiality of governance. The ISO 15118 connection between EV and CP uses Transport Layer Security (TLS) with unilateral server-side authentication of the CP, as required for PnC charging [28].[1] The OCPP and OCPI connections use TLS with bilateral authentication respectively.

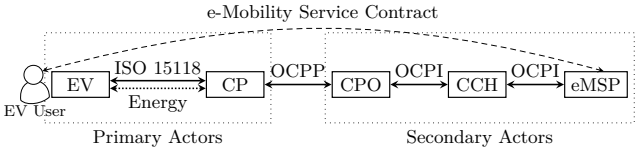

Fig. 1. EV Charging System Model

2.1 Credential Provisioning

For the installation of PnC contract credentials into an EV, ISO 15118 defines a credential provisioning process. EVs are initially provided with provisioning credentials by the EV's Original Equipment Manufacturer (OEM), i.e., the vehicle manufacturer. An EV user can use the unique Provisioning Certificate Identifier (PCID) to register the EV at an eMSP after the conclusion of a charging contract. During the first charging session, the EV sends a request for contract credentials to the CP. This request includes the provisioning certificate and is signed with the corresponding private key. The request is forwarded from the CP over the CPO to the CCH (acting as Certificate Provisioning Service (CPS)).[2]

The CCH verifies the signed request and forwards it to the eMSP, who can identify the corresponding user based on the registered PCID. The eMSP generates new contract credentials, which are uniquely identified by an e-Mobility Account Identifier (eMAID) included in the contract certificate, and encrypts the private contract key with the public provisioning key from the provisioning certificate. Afterwards, the contract credentials are sent to the CCH who verifies this data and creates a signature over it. The signed contract credential data is forwarded over the CPO and the CP to the EV. The EV verifies the signature over the contract credentials and saves them for later use.

Credential Pools: While the exact backend interactions are not defined in ISO 15118, the DKE application guideline in [59] presents a method based on certificate

1 For the CP's certificate (and all other ISO 15118-related certificates), the standard specifies its own Public Key Infrastructure, which defines certificate profiles and roots of trust [28].

2 A CPS is a trusted intermediary for credential installation (cf. [28], Section 7.9.2.5). For sake of simplicity, we assume that the CCH implements CPS functionality.

pools that allows the generation of contract credentials to run independently of an EV's credential request. [59] introduces two certificate pools: a provisioning certificate pool and a contract certificate pool. For simplicity, we assume both pools are implemented by the CCH. The provisioning certificate pool is used by OEMs to store provisioning certificates of EVs such that eMSPs can access them after an EV user registers their PCID with a charging contract. Hence, eMSPs can also generate contract credentials (with encrypted private key) directly after EV user registration, independently of the EV's contract credential request. The pre-generated contract credentials (together with the corresponding PCID) are uploaded into the contract certificate pool such that a CCH can directly retrieve and sign them after receiving an EV's contract credential request.

2.2 Charge Authorization and Billing

After an EV is provisioned with contract credentials, it can authenticate itself at a CP using PnC authentication, which implements a challenge-response protocol between EV and CP. In order to authorize a charge session, an additional validation of the contract credential's eMAID via the backend is possible. After a charge session is completed, metering information of the session needs to be sent to the backend to bill the EV user. The following describes these processes in detail.

First, for PnC authentication, the EV sends its eMAID and contract certificate chain, to the CP. The CP can request the revocation status of the contract certificate (chain) from its CPO. Alternatively, the CP could request a full certificate path validation from its CPO (instead of only revocation information).

After the EV's certificate chain is validated, the CP sends a random nonce to the EV. The EV responds with a signature over the nonce (created with the private contract key) and the CP verifies this signature using the public key from the EV's contract certificate. A successful authentication of the EV could be used as implicit authorization for the user to charge (cf. [27], Use case D1 and [43], Requirement C07.FR.12). Alternatively, an explicit authorization of the contract certificate's eMAID might be required (cf. [27], Use case D2).

The CP can request the charge authorization status of the eMAID from its CPO. The CPO can then forward the request via the CCH to the eMSP. The eMSP verifies the user's authorization status and sends a response on the reverse path. Alternatively, for a faster authorization of eMAIDs, an eMSP may distribute a list of authorized IDs over the CCH to CPOs. Afterwards, the CCH and respective CPOs can respond to authorization requests independent of the eMSP. Additionally, a CPO may distribute a list of authorized IDs to its CPs, allowing for offline charge authorization. Similar to charge authorization, CP reservation is possible by registering a user's eMAID at a CP for a specific time period.

While the EV is charging, the CP periodically queries its electricity meter. The resulting meter information is periodically sent to the EV. Optionally, the CP can request a signature over the meter values from the EV (i.e., a signature from the EV with its private contract key; called *metering receipt* in ISO 15118). Additionally, the CP sends the meter values to the CPO either during the session or later on (e.g., if the CP is offline during the charging session).

For the billing of charging sessions, the CPO generates a Charge Detail Record (CDR) for the session based on the transaction information it received from its CP. The CPO sends this CDR over the CCH to the eMSP. The CDR contains all billing relevant information, including, start and end date of the session, the user's eMAID, location information, applicable tariffs, consumption information, and optional signatures.

2.3 Personal Data

The personal data that is received/known by the different actors in the EV charging infrastructure for the processes of credential provisioning, charge authorization, and billing is shown in Table 1. Personal data that is arguably not required for the respective actors' operation – and thus should be omitted for compliance with the General Data Protection Regulation's data minimization provision – is marked in brackets. We only consider data that is involved in the respective protocols and any external personal data (e.g., from cameras at a CPO's charging site) is out of scope.

During credential provisioning, every actor can receive the EV's provisioning certificate (including PCID), which uniquely identifies the EV. While the CP and CPO simply forward this data (i.e., do not need to know its contents), the CCH or eMSP require this data in order to respectively either retrieve the corresponding response from the contract certificate pool or to generate the response. Similarly, the contract credentials (including eMAID), which can be used to identify an EV user, simply need to be forwarded by CP and CPO. The eMSP, however, generates the contract credentials and thus always knows this data. Depending on the trust

Table 1. Personal Data in the EV Charging Infrastructure

	CP	CPO	CCH	eMSP
Credential Provisioning				
Provisioning Certificate	(X)	(X)	X	X
Contract Credentials	(X)	(X)	(X)	X
Charge Authorization				
eMAID	(X)	(X)	(X)	X
Location	X	X	(X)	(X)
Time	X	X	X	X
Charge Billing (CDRs)				
eMAID	(X)	(X)	(X)	X
Location	X	X	(X)	(X)
Transaction Information	X	X	(X)	X

X = actor knows this personal data; arguably required for operation
(X) = actor knows this personal data; arguably not required for operation

relationship and used security measures between eMSP and CCH, the CCH (acting as CPS) might not need to know any personal data from the contract credentials in order to sign the credential installation response.

During charge authorization, every actor can receive the EV user's eMAID along with the charge location (i.e., the CP) and time. For offline authorization, only the CP would receive this data. This combination of data is especially privacy-relevant as it allows the generation of detailed movement profiles for an EV user by linking their different charge locations via the static eMAID. With regard to data minimization, only CP/CPO need to know the location (a CPO is assumed to know where its CPs are). Additionally, every actor presumably knows the current time independently of received data and is thus able to relate it to a charge authorization. For an online authorization request via the eMSP, only this eMSP would need to receive the user's eMAID. However, to enable the CCH or CPO to respond to authorization requests based on an eMSP's distributed list of authorized IDs and to enable offline authorization, the CP, CPO, and CCH also need to receive the user's eMAID. Thus, in order to prevent the generation of movement profiles without restricting functionality, linkability of the user's authorization data should be avoided.

For the billing of charge sessions, every actor receives the EV user's eMAID along with the charge location and additional transaction information (time, duration, consumption, etc.). With regard to data minimization, the eMSP needs to know the eMAID along with the transaction information to bill the user for the charging session. Similarly to charge authorization, the CPO is assumed to know the location of its CPs. Additionally, the CPO needs to know the transaction information in order to bill the eMSP for their user's charging session.

3 Security, Privacy, and Functional Requirements

For the identification of security and privacy requirements, we analyzed the EV charging system model for threats using STRIDE [22] and LINDDUN [10]. STRIDE is often regarded as the most mature and widely used security threat modeling methodology [25, 48]. Additionally, LINDDUN is one of the most well-known Data Protection Impact Assessment frameworks [4] and offers a methodological approach along with an extensive privacy knowledge base for the systematic elicitation of privacy threats, whereby it partially shares the same methodological steps as STRIDE [49, 63]. For our threat analysis, we consider the following adversaries with relevance for EV charging:

Network Hacker: The adversary can eavesdrop, modify, inject, or drop any messages in the network but cannot break cryptography as in the Dolev-Yao model [12]. As a Man-in-the-Middle, the adversary can try to charge their EV on the cost of another user, e.g., by using replay attacks. By eavesdropping on the communication channel, the adversary tries to collect personal data of EV users in order to build detailed movement profiles of the users (e.g., to rob a user's home in their absence [15, 24]).

Local Adversary: As a purely remote adversary is often considered to be too weak for comparable settings [1, 11, 33, 41, 45], we also consider a Local Adversary with physical access to any unattended EV and CP in public areas. The adversary can physically tamper with the system and extract or modify any data stored on the system, e.g., steal private keys, read out charging logs, or install a malicious firmware image (cf. [8] for such attacks on CPs). The goal of the adversary could be the collection of personal data of EV users and/or to charge their EV at another user's cost. However, the adversary cannot access data stored in tamper-protected areas (e.g., a TPM).

Curious Operator: While the potential of ill-intentioned operators raises serious privacy concerns [38], modeling these operators as active adversaries is generally too strong as they are restricted by regulations and the desire to maintain reputation [44]. Hence, we consider the legitimate operators of backend systems as honest-but-curious. The Curious Operator does not deviate from the defined protocol but attempts to learn as much information

as possible from legitimately received messages [44]. Additionally, a collusion between different actors is excluded (e.g., due to regulatory restrictions). Operators, for instance, could try to build movement profiles of EV users and/or sell information to other companies for targeted advertising [34].

According to the used threat modeling methodologies and under consideration of the described adversaries, we modeled the system model from Section 2 via Data Flow Diagrams and identified the relevant threats. Details can be found in [30]. In summary, EV users are subject to spoofing threats as their credentials are insufficiently protected from a Local Adversary. Additionally, every backend actor receives enough linkable data (long-term pseudonyms, charging locations, etc.) to build detailed movement profiles of EV users. Based on the full list of identified threats, we define security- (SR_x) and privacy requirements (PR_x) for a solution to provide a secure and privacy-preserving charging and billing of EVs as follows:

SR_1 *Secure Credential Storage/Usage:* Credentials stored on the EV should be protected during storage and usage. Access to the credentials should be conditioned on the integrity of the EV's software state.

SR_2 *Secure Credential Installation:* Bilateral authentication of the data sent between CCH (acting as CPS) and EV for credential installation as well as the secrecy of involved credentials should be guaranteed.

SR_3 *Secure Charge Authorization:* A solution should guarantee the eMSP that charge authorization requests from an EV are authentic.

SR_4 *Charge Data Authenticity:* A solution should guarantee the eMSP the authenticity of received charge data as attested by the EV[3] as well as an authentic binding between authorization and charge data.

PR_1 *Data Minimization:* Actors should only receive minimal personal data.

PR_2 *Unlinkable Credential Installation:* Multiple credential installations should not be linkable to an EV by anyone but eMSP and CCH (acting as CPS).

PR_3 *Unlinkable Charge Authorization:* Multiple authorizations of an EV user should not be linkable using the authorization data by anyone but the eMSP.

PR_4 *Unlinkable CDRs:* The billing relevant data of multiple charge sessions should not be linkable to the same EV (user) by any actor but the eMSP.

PR_5 *Unlinkability of EV Users and Locations:* No actor should be able to link an EV user and their charge locations.

In order to guarantee the usability of the solution, we identified the following functional requirements (FR_x) that must be fulfilled:

FR_1 *Low Overhead:* The additional overhead of a solution should be low and not cause issues with the timing/size constraints of existing standards.

FR_2 *Minor Changes:* It should be possible to integrate the solution into the existing charging protocols with only minor changes.

FR_3 *Full Feature Support:* Features and services of current EV charging protocols should be supported to the largest extend possible.

4 Privacy-Preserving PnC Extension

To address the requirements from Section 3, we propose an extension for protocols of the PnC charging architecture to enable privacy-preserving charge authorization and billing. We use a TPM in an EV for protecting ISO 15118 credentials similar to [19] and extend this solution with a TPM-based DAA scheme [61] (adapted to be compatible with the PnC architecture) to provide unlinkability of personal data whenever possible. Notably, the use of a TPM is a suitable option in this context, since TPMs have already seen adoption in the automotive industry [26] and moreover the current draft of the next version of ISO 15118 [29] already offers explicit support for a TPM (based on the concept of [16]). Our extension can be easily integrated into existing protocols and remains compatible with the general procedures and data flows of current EV charging protocols.

The basic idea of our privacy-extension is as follows. The TPM is used for secure storage and usage of credentials stored in the EV and ensures these credentials can only be used if the respective control unit is in a trustworthy state. A DAA key pair is generated by the TPM and the eMSP issues a DAA credential for the public DAA key of the EV. Afterwards, the EV can use these DAA credentials to certify TPM-generated session key pairs (usable for authentication during a charge session

3 While billing relevant data may originate from different sources (e.g., meter values from the CP), verifying the correctness of this data on EV side is out of scope.

Table 2. TPM 2.0 Key Templates

	EC	PC	CC^{DAA}	CC^{Sess}	SK_{eMAID}
type:	TPM2_ALG_ECC				TPM2_ALG_ KEYEDHASH
nameAlg:	TPM2_ALG_SHA256				
object-Attributes:	*fixedTPM*, *fixedParent*, *restricted*, *decrypt*, *adminWithPolicy*, *sensitiveDataOrigin*	*fixedTPM*, *fixedParent*, *sign*, *decrypt*, *sensitiveDataOrigin*	*fixedTPM*, *fixedParent*, *restricted*, *sign*, *sensitiveDataOrigin*	*fixedTPM*, *fixedParent*, *sign*, *userWithAuth*, *sensitiveDataOrigin*	*sign*, *userWithAuth*
authPolicy:	TPM2_PolicySecret(TPM_ RH_ENDORSEMENT)	$Pol_{Auth} =$ TPM2_PolicyAuthorize(OEM_{pk})		$\perp$	
symmetric:	AES-128-CFB	n/a			
scheme:	n/a		TPM2_ALG_ECDAA TPM2_ALG_SHA256	$\perp$	TPM2_ALG_HMAC TPM2_ALG_SHA256
curveID:	TPM2_ECC_NIST_P256		TPM2_ECC_BN_P256	TPM2_ECC_NIST_P256	n/a

towards a CP). Additionally, the EV's TPM receives a symmetric key from the eMSP for the secure generation of authorization values that are unlinkable by anyone but the eMSP (for billing of the user).

In Section 4.1, we describe the different types of keys stored in the TPM and in Section 4.2 the required certificate extension for transporting keys and policies. We describe the initialization of EV and other actors in Section 4.3 before we present our adaptions of the PnC processes in Sections 4.4 and 4.5.

4.1 TPM 2.0 Key Profiles

The TPM specification allows the definition of attributes and policies for each TPM key. Attributes determine how the TPM may use a key (e.g., only for signing) and a policy can define specific tests that have to pass before key usage is authorized [54].

Our privacy-extension requires five different types of TPM keys (cf. Table 2): (i) an endorsement credential (EC) key pair following the Trusted Computing Group's endorsement credential profile [56], (ii) a provisioning credential (PC) key pair following [19], (iii) a DAA contract credential (CC^{DAA}) key pair following that of the DAA key in [61] with the addition of an authorization policy, (iv) a session contract credential (CC^{Sess}) key pair (generated anew for every communication session), and (v) a symmetric eMAID key (SK_{eMAID}) defined for the generation of Hash-based Message Authentication Code (HMAC)-based charge authorization values.

All key pairs are used for Elliptic Curve Cryptography (ECC) whereas the eMAID key is an HMAC key (KEYEDHASH). The *fixedTPM* and *fixedParent* attributes indicate that a key cannot be exported out of the TPM or moved within its hierarchy (cf. [54], Section 25). The *sensitiveDataOrigin* attribute indicates that private keys were generated by the TPM. The *decrypt* and *sign* attributes indicate the usages of the keys and the *restricted* attribute limits key usage to certain commands and on specifically formatted objects. The PC and CC^{DAA} key pairs require policy authorization based on the same *authPolicy* value Pol_{Auth} (a `TPM2_PolicyAuthorize` with the OEM's public key OEM_{pk}) such that credential provisioning and charge authorization require an assertion of a separate OEM-signed policy (verifiable with OEM_{pk}). The *curveID* field indicates an ECC key pair's curve, i.e., BN_P256 for the CC^{DAA} key pair and otherwise NIST_P256.[4]

4.2 Certificate Extensions

Since the security of DAA credentials is provided by the endorsement EC key pair in the EV's TPM, eMSPs need to trust in the authenticity of the respective public key EC_{pk} when they issue DAA contract credentials. Additionally, eMSPs need to receive the intended *authPolicy* of a CC^{DAA} key pair in a trustworthy manner for the TPM's credential protection mechanism. Using the method from [19], this trust can be provided by an inclusion of EC_{pk} and the intended *authPolicy* in the EV's provisioning certificate PC_{cert}, which is signed by the OEM and verifiable by eMSPs, via a non-critical Subject Information Access (SIA) extension as shown in Fig. 2a. Additionally, in order to provide an eMSP's

4 While the BN_P256 curve provides less than 128-bit security [3] and the TPM specification includes the BN_P638 curve [55] (with more than 128-bit security [47]), most TPMs do not support BN_P638 yet [61] and BN_P256 is still used here.

OEM Provisioning Certificate (PC_{cert})		
Version:		X.509v3 (0x2)
Serial Number:		12345 (0x3039)
Signature Algorithm:		ecdsa-with-SHA256
(no changes to certificate profile form ISO 15118; cf. [28], Annex F)		
X509v3 Extensions	SIA:nc	**OID:1.0.20.4**
		TPM Public EC Key (EC_{pk})
		512 bit OCTET STRING in Base64
		OID:1.0.20.5
		TPM SHA256 Policy Digest (Pol_{Auth})
		256 bit OCTET STRING in Base64
Signature	Algorithm:	ecdsa-with-SHA256
	Value:	OCTET STRING (Signature from Issuer)

(a) Provisioning Certificate (cf. [19])

eMSP Certificate ($eMSP^{DAA}_{cert}$)		
Version:		X.509v3 (0x2)
Serial Number:		12345 (0x3039)
Signature Algorithm:		ecdsa-with-SHA256
Issuer:		CN=eMSPSubCA1, O=ISO
Validity	Not Before:	May 7 08:40:32 2020 GMT
	Not After:	May 6 08:40:32 2050 GMT
(no changes to certificate profile form ISO 15118; cf. [28], Annex F)		
X509v3 Extensions	SIA:nc	**OID:1.0.20.6**
		Public DAA Group Key ($eMSP^{DAA}_{pk}$)
		2048 bit OCTET STRING in Base64
Signature	Algorithm:	ecdsa-with-SHA256
	Value:	OCTET STRING (Signature from Issuer)

(b) eMSP Certificate with SIA Extension

Fig. 2. Extended ISO 15118 Certificate Profiles

public group key $eMSP^{DAA}_{pk}$ to verifiers in an authentic manner, we include this key in the eMSP's certificate via a non-critical SIA extension as shown in Fig. 2b.

4.3 Preparations

In order to set up our privacy-extension, the EV and its TPM need to be initialized, the CPs, CPOs, the CCH, and eMSPs need to be prepared for the handling of certificate installation requests, and the DAA scheme needs to be set up.

Initialization of the EV: EV initialization requires the steps for preparation of a TPM for ISO 15118 use from [19]; i.e., the OEM instructs the EV's TPM to generate the EC and PC key pairs some time before the initial delivery. The OEM reads out the respective public keys, i.e., EC_{pk} and PC_{pk}, and generates the provisioning certificate PC_{cert} for PC_{pk} including EC_{pk} and Pol_{Auth} (i.e., a `TPM2_PolicyAuthorize` for the OEM's public key OEM_{pk}; cf. Section 4.1). If certificate pools are used, DAA contract key pairs $(CC^{DAA}_{pk}, CC^{DAA}_{sk})$ can already be generated by the TPM and the public keys can be uploaded by the OEM together with PC_{cert} and the PCID into the provisioning certificate pool. The OEM provides a `TPM2_PolicyPCR` digest (a policy conditioned on the EV's software state) and signs it in order to authorize its use with Pol_{Auth}.

Initialization of Other Actors: In order to forward an EV's certificate installation request, eMSPs inform the CCH about newly registered PCIDs, and the CCH stores the PCID to eMSP mappings. If certificate pools are used, the CCH uses the PCID to find an EV's credential in the pool. Additionally, to protect the personal information in certificate installation requests, the requests must be encrypted by the EV for the CCH. For this, the CPO distributes the public key certificate chains of relevant CCHs to its CPs. Furthermore, the DAA scheme requires the general setup from [61]. The eMSPs' key pairs ($eMSP^{DAA}_{pk}$, $eMSP^{DAA}_{sk}$) are generated and the respective public keys are included in an eMSP's certificate $eMSP^{DAA}_{cert}$ (cf. Fig. 2b) such that they can be validated by CPs, CPOs, and the CCH based on an eMSP certificate root $eMSP_{Root}$.

Offline Authorization: In order to enable offline authorization at CPs, eMSPs can pre-generate authorization values $\langle \mathcal{M}^i_{id}, \mathcal{M}^i_{auth} \rangle$ for their users, whereby $\mathcal{M}^i_{id} = Hash(hmac_{SK_{eMAID}}(00||i))$ and $\mathcal{M}^i_{auth} = hmac_{SK_{eMAID}}(01||i)$, for each user's SK_{eMAID} and each authorization number i in $[1, \ldots, n]$. For unlinkability, authorization values cannot be reused and need to be regularly generated and distributed. The eMSP sends a shuffled list of authorization values over the CCH to all applicable CPOs. To prevent replay of authorizations at a different CP and the spoofing of an EV user after information leakage at a CP, the CPO transforms every entry's $\mathcal{M}^i_{id}$ into $CP\mathcal{M}^i_{id} = Hash(\mathcal{M}^i_{id}||CSID)$ and $\mathcal{M}^i_{auth}$ into $CP\mathcal{M}^i_{auth} = Hash^2(\mathcal{M}^i_{auth}||nonce^i_x)$, whereby $CSID$ uniquely identifies the target CP, $nonce^i_x$ is a random nonce (preventing reuse), and $Hash^n(m) = Hash^{n-1}(Hash(m))$ with $Hash^1(m) = Hash(m)$. Hence, the CPO's whitelist for a CP contains the tuples: $\langle CP\mathcal{M}^i_{id}, nonce^i_x, CP\mathcal{M}^i_{id} \rangle$. Similar to offline authorization, CP reservation is also possible via the pre-generation of authorization values.

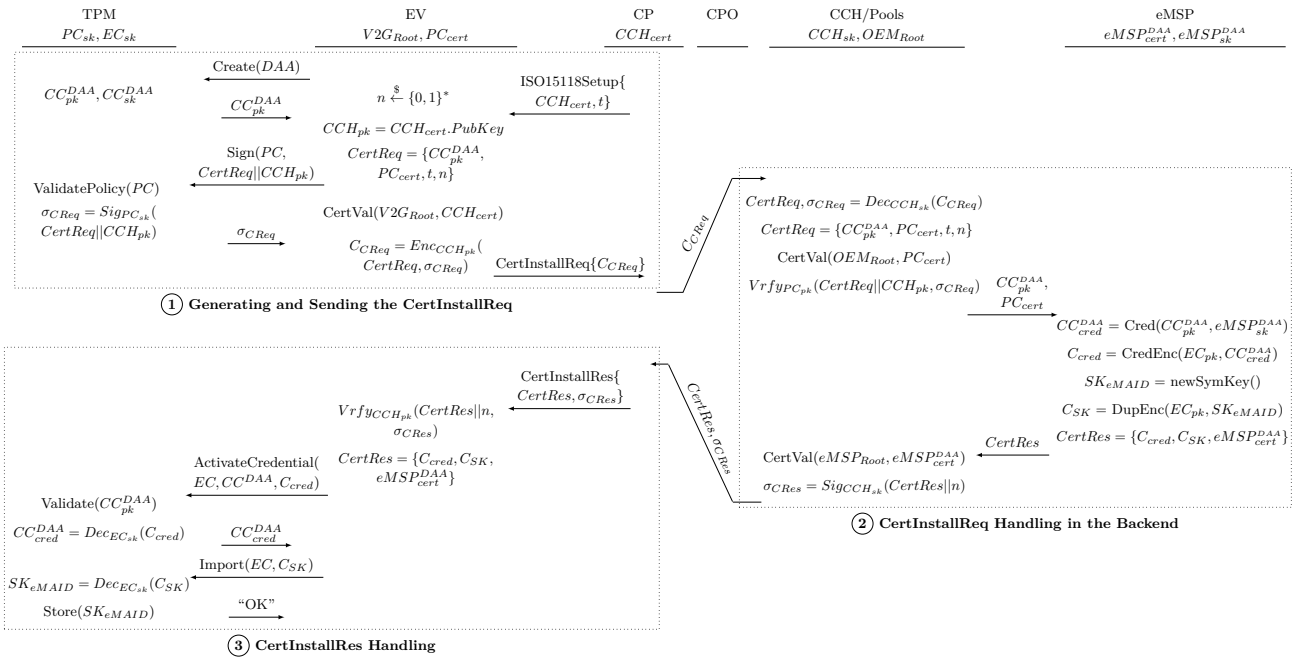

Fig. 3. Contract Credential Installation Procedure

4.4 Contract Credential Provisioning

The provisioning process starts with the generation of a certificate installation request as shown in Fig. 3, Part 1. Hereby, the base DAA scheme [61] is adapted to support the characteristics of the PnC architecture, such as (i) no preexisting trust in the eMSP's (issuer's) public key by EVs before credential installation, (ii) use of certificate pools (i.e., possibly no live interaction with the credential issuer), and (iii) privacy threats during credential installation.

The EV starts by generating a DAA contract key pair $(CC_{pk}^{DAA}, CC_{sk}^{DAA})$ in its TPM. Over ISO 15118, the EV receives a timestamp t (used for freshness of the request without requiring trust in the CP) and the CCH's certificate chain (CCH_{cert} for public key CCH_{pk}) from the CP.

Afterwards, the EV can use its new CC_{pk}^{DAA} to build the certificate request data $CertReq$ including its PC_{cert}, the timestamp t, and a fresh nonce n (used for freshness of the following response). The EV instructs its TPM to sign $CertReq||CCH_{pk}$ with the private provisioning key PC_{sk}, which requires an assertion of the OEM-signed `TPM2_PolicyPCR` (cf. Section 4.3), i.e., it requires that the EV booted into a trusted software state. The signed request is encrypted for the CCH and sent to the CP, which forwards it over the CPO to the CCH.

The CCH decrypts the request, validates PC_{cert}, and verifies the signature over and freshness of $CertReq$ (cf. Fig. 3, Part 2). If certificate pools are used (not shown), the CCH loads the preexisting certificate installation responses ($CertRes$). Otherwise, the CCH forwards the request to the eMSP.

The eMSP generates the DAA credential CC_{cred}^{DAA} for CC_{pk}^{DAA} as detailed in [61]. Afterwards, the eMSP encrypts CC_{cred}^{DAA} with the public endorsement key EC_{pk} from the extension in the EV's PC_{cert} using the TPM credential protection mechanism, which ensures that decryption is only possible if CC_{sk}^{DAA} belongs to the same TPM as EC_{sk} (cf. [54], Section 24). Additionally, the eMSP generates the symmetric key SK_{eMAID} (used to uniquely identify the EV user's account) and encrypts SK_{eMAID} with EC_{pk} using the TPM's object duplication procedure (cf. [54], Section 23.3), such that decryption is only possible via an import into the TPM. Finally, the eMSP builds the certificate installation response $CertRes$ with the generated data as well as its own certificate $eMSP_{cert}^{DAA}$. $CertRes$ is sent to the CCH, which validates the eMSP's certificate $eMSP_{cert}^{DAA}$. The CCH signs $CertRes$ together with the EV's nonce and forwards the signed $CertRes$ over the CPO to the CP.

The CP forwards the received data to the EV over ISO 15118 (cf. Fig. 3, Part 3). Afterwards, the EV verifies the signature over as well as the freshness of $CertRes$, decrypts the DAA credential CC_{cred}^{DAA} with its TPM, and decrypts SK_{eMAID} by importing it into the TPM.

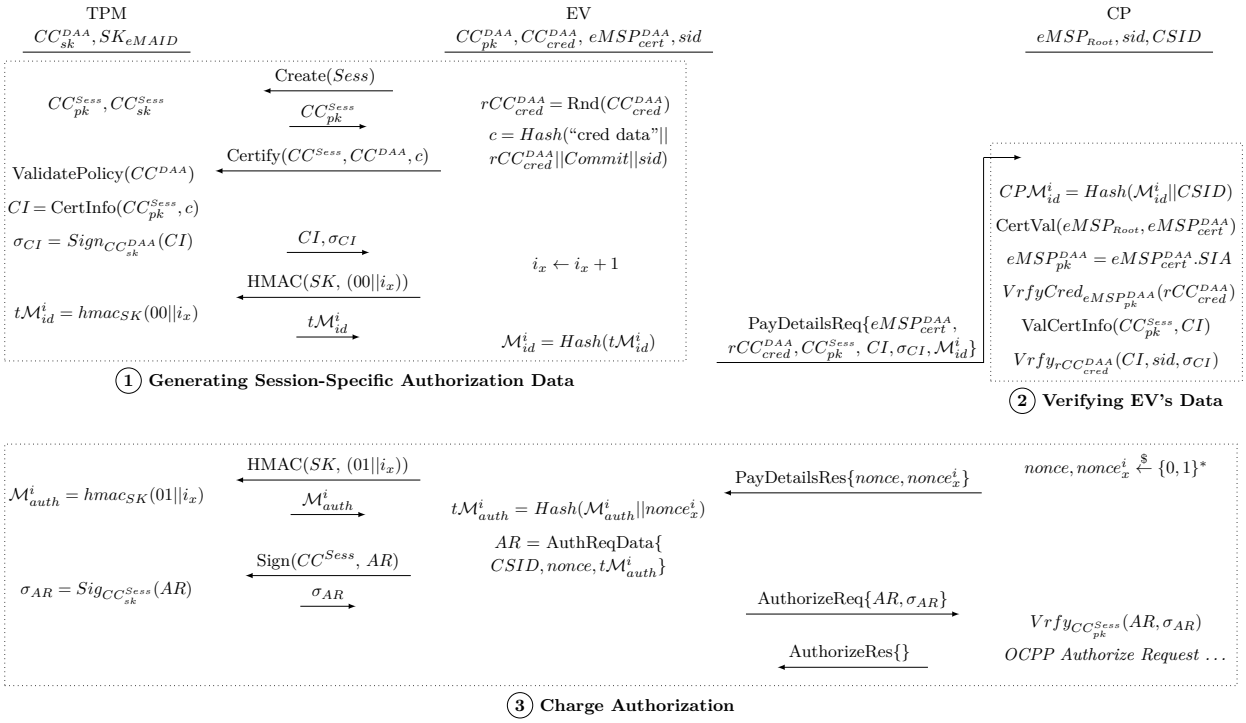

Fig. 4. Charge Authorization

4.5 Charge Authorization and Billing

The process for privacy-preserving PnC authorization uses DAA-based authentication of a session key via the certify protocol from [61] (with an additional binding to the ISO 15118 session) and uses SK_{eMAID} as shared secret between EV and eMSP to generate pseudonyms that are unlinkable by third parties.

The EV starts by instructing its TPM to create the new session key pair ($CC^{Sess}_{pk}, CC^{Sess}_{sk}$; cf. Fig. 4, Part 1) and randomizes its DAA contract credentials CC^{DAA}_{cred} into a unique rCC^{DAA}_{cred} (still verifiable using $eMSP^{DAA}_{pk}$; cf. [61]). Afterwards, the EV uses its TPM to certify the session key pair with CC^{DAA}_{sk}, i.e., to create a DAA signature over the session key's public values including the qualifying data c as defined in [61]. Since only the DAA key pair is sealed to the EV's software integrity (via `TPM2_PolicyPCR`), the certification is additionally bound to the ISO 15118 session via c such that each charge requires an assertion of `TPM2_PolicyPCR`. The certify information data CI and the generated signature σ_{CI} are returned to the EV. The EV continues by generating $\mathcal{M}^i_{id} = Hash(hmac_{SK_{eMAID}}(00||i))$, using its TPM for the HMAC calculation and its current authorization counter value i. Afterwards, the EV sends $eMSP^{DAA}_{cert}, rCC^{DAA}_{cred}, CC^{Sess}_{pk}, CI, \sigma_{CI}$, and $\mathcal{M}^i_{id}$ in its *PayDetailsReq* message to the CP (instead of the usual eMAID and contract certificate). The CP verifies $eMSP^{DAA}_{cert}$ and rCC^{DAA}_{cred}, validates that CC^{Sess}_{pk} corresponds to CI, and verifies the DAA signature σ_{CI} over CI using rCC^{DAA}_{cred} (cf. Fig. 4, Part 2). Note that a CPO may handle these validations for CPs (e.g., if a CP does not support DAA).

If all validations are successful, the CP generates a fresh $nonce^i_x$ (or uses $\mathcal{M}^i_{id}$ to find $nonce^i_x$ in its authorization whitelist for offline authorization) and sends $nonce^i_x$ together with the usual ISO 15118 PnC *nonce* to the EV in a *PayDetailsRes* message (cf. Fig. 4, Part 3). The EV uses $nonce^i_x$ to build $t\mathcal{M}^i_{auth} = Hash(\mathcal{M}^i_{auth}||nonce^i_x)$ with $\mathcal{M}^i_{auth} = hmac_{SK_{eMAID}}(01||i)$, again using its TPM for the HMAC calculation. Additionally, the EV signs *AuthorizeReq*$\{CSID, nonce, t\mathcal{M}^i_{auth}\}$ with CC^{Sess}_{sk} via its TPM – whereby $CSID$ is a unique identifier of the CP's certificate during the TLS handshake (not shown) – and sends the signed message to the CP. After receiving the *AuthorizeReq* message, the CP verifies the signature using the previously received (and verified) CC^{Sess}_{pk}. Finally, the CP verifies the EV user's authorization to charge either using its authorization whitelist (i.e., $Hash(t\mathcal{M}^i_{auth}) \stackrel{?}{=} CP\mathcal{M}^i_{auth}$) or a request to the backend for the user's authorization data (i.e., $\mathcal{M}^i_{id}$, ID_{eMSP}, $t\mathcal{M}^i_{auth}$, and $nonce^i_x$) whereby information on

the charge location is omitted in messages to CCH and eMSP. Later during the ISO 15118 session, the EV can sign meter receipts and other billing-relevant data using CC_{sk}^{Sess} via its TPM. For a secure binding between EV user authorization and billing data authenticity, the billing data is appended with $EV_h = Hash(\text{“EVh”}||\mathcal{M}_{auth}^{i}||CC_{pk}^{Sess})$ before signing.

For the billing of the charging session, the CPO is still informed about the user's consumption and the user's eMSP. Thus, the CPO can still bill the eMSP for their users' consumed energy. The eMSP on the other hand is informed about the user's consumption and can identify the user based on the used authorization data. Thus, an eMSP is still able to bill its users for their charging sessions. Note that in order to fully avoid the unnecessary dissemination of personal data (cf. Table 1), all billing-relevant transaction data could be encrypted by the CPO for the eMSP. However, since the CCH cannot link this data to the charge location or a static user ID, it is arguably already de-identified enough to thwart privacy-risks even without encryption.

5 Implementation and Functional Evaluation

We implemented our privacy-extension as a proof-of-concept. The setup is shown in Fig. 5. EV and CP are based on Raspberry Pi 3 Model B boards and connected via PLC stamp micro 2 EVBs in order to emulate Power Line Communication in accordance with ISO 15118. The EV Pi is equipped with a hardware Infineon SLM 9670 TPM 2.0. CPO, CCH, and eMSP are implemented as Virtual Machines based on Oracle VirtualBox. The CP Pi uses its WLAN uplink for communication with the backend.

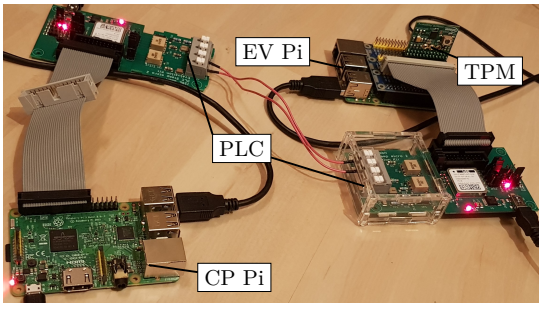

Fig. 5. Proof-of-Concept Setup

The ISO 15118 implementation uses RISE V2G [57] in Java, the OCPP 2.0.1 implementation is based on a Python OCPP framework [51], and the OCPI implementation uses web services in Python. The implementation of all DAA calculations that do not involve the TPM (verification of DAA signatures, etc.) are based on the C++ open-source implementation from [61] (cf. [60]). The EV Pi's interactions with its TPM are implemented in C with the TPM2-TSS [53] and run in parallel to the ISO 15118 process in order to pre-calculate steps whenever possible and represent an optimized integration.

We use the implementation to evaluate the resulting overhead in comparison to the existing methods of the EV charging protocols. The largest communication-/storage overhead is created if whitelist-based offline authorization is supported as new values need to be distributed regularly. Assuming a scenario with 277,777 EV users per eMSP and EV's being charged once a day on average (cf. [50]), the weekly whitelists per eMSP in the backend would be 124.4 MB and at CPs would be 155.5 MB. The overhead of the default (not privacy-preserving) whitelist approach is in comparison arguably negligible as new values only need to be distributed for changes (e.g., for an eMSP's new EV users). However, compared to the general communication overhead in the e-mobility infrastructure, the whitelist overhead of the proposed solution can still be judged as relatively low. For instance, an EV's *default* (non-DAA) PnC authentication alone already requires the transfer of $1,811$ bytes (including certificate chain, nonce, and signature; based on the reference RISE V2G implementation), which equals 3521.3 MB weekly assuming the same scenario (277,777 EV users and one charge a day).

Additional communication overhead is shown by the increase in ISO 15118 message sizes, namely *CertInstallReq* from 811 to 952 bytes (17.39% increase), *CertInstallRes* from 3637 to 4633 bytes (27.39% increase), *PayDetailsReq* from 1452 to 1973 bytes (35.88% increase), *PayDetailsRes* from 51 to 55 bytes (7.84% increase), and *AuthorizeReq* from 308 to 355 bytes (15.26% increase).

Notably, the evaluation of communication overhead shows a potential problem with regard to compliance to size constraints in existing standards. Namely, the eMSP's certificate with SIA extension was 863 bytes while ISO 15118 limits the size of certificates to 800 bytes (cf. [28], Requirement V2G2-010). However, in order to address the eMSP's certificate size, the public group key could be included via a custom extension (compliant to RFC 5280; cf. [9], Section 4.2) that only consists of an OID and the public group key in byte form, which results in a 755 byte certificate.

Table 3. Measured Computational Overhead for the Normal and Proposed Methods

Process/Steps	Time in ms				ISO 15118
	Mean	σ	Max	Min	Time Limit in ms
Normal Contract Credential Provisioning					
CertInstallReq Generation	1596.9	±23.3	1742.2	1569.2	40000
CertInstallReq Handling	409.2	±20.1	457.9	361.2	4500
CertInstallRes Handling	1685.7	±62.9	1831.1	1603.6	40000
Privacy-Preserving Contract Credential Provisioning					
CertInstallReq Generation	1858.8	±124.1	2880.8	1805.2	40000
CertInstallReq Handling	341.9	±30.2	440.3	293.8	4500
CertInstallRes Handling	717.1	±24.7	760.4	663.0	40000
Normal Charge Authorization					
PayDetailsReq Generation	24.3	±50.7	138.6	1.0	40000
PayDetailsReq Handling	94.1	±58.8	220.5	54.9	4500
AuthorizeReq Generation	73.3	±9.8	96.9	41.0	40000
Offline Authorization Validation	67.3	±6.6	85.9	56.3	1500
Privacy-Preserving Charge Authorization					
PayDetailsReq Generation (after install)	1063.4	±32.5	1126.7	960.6	40000
PayDetailsReq Generation (not after install)	9.9	±1.9	16.2	7.1	40000
PayDetailsReq Handling	322.5	±18.7	363.8	282.2	4500
AuthorizeReq Generation	129.2	±4.9	150.8	118.2	40000
Offline Authorization Validation	79.0	±10.7	111.4	62.3	1500

In order to evaluate the computational overhead of our privacy-extension, times were measured in the EV Pi's ISO 15118 implementation (as this best reflects the impact on the user) using Java's *System.nanoTime()*. We repeated all measurements 100 times and list the averages per high-level process. Additionally, Table 3 lists the measured average times per process step along with the standard deviation (σ) as well as min and max values. Table 3 also includes the respective ISO 15118 time limits (cf. [28], Section 8.7.2) since, of the considered protocols, ISO 15118 is the only protocol with specific timing constraints.

The total time for our credential provisioning was 2917.8 ms, which is 774 ms faster than the reference implementation time of 3691.8 ms[5] (20.97% decrease). The time for our charge authorization directly after credential installation was 1594.1 ms. The time for authorization in following sessions (i.e., the more common case with already installed credentials) was 540.6 ms due to possible pre-calculations. Specifically, possible pre-calculations include the loading of the required keys into the TPM, the creation of a session key pair in the TPM, the randomization of the DAA credential, and the calculation of TPM policy sessions during the communication setup between EV and CP. Additionally, the execution of the TPM certify command can be started as soon as the EV receives the ISO 15118 session ID in the header of the CP's session setup message and is finished before the *PayDetailsReq* message can be sent. Compared to the time of the reference implementation of 259 ms, the added overhead is relatively low with an increase of 1335.1 ms without pre-calculations (515.48% increase) or a more common 281.6 ms with pre-calculations (108.72% increase). Especially considering that the prior *default* ISO 15118 session setup alone takes 3305.3 ms (±25.3 ms standard deviation), the perceived additional overhead can be considered low (from EV plug-in until the start of charge there is only a 7.9% increase in the case with pre-calculations). The times for *AuthorizeReq* generation with both the default and proposed method (cf. Table 3) are also representative of the overhead for authenticity-protecting the billing-relevant data (i.e., the meter receipts) since in both cases both processes involve the same cryptographic actions (hashing and a signature with the (session) contract key).

5 The relatively long time of the reference implementation is mostly caused by interactions with Java *KeyStores* in the RISE V2G implementation.

The functional evaluation shows that our extension can be integrated into existing protocols with only minor overhead FR_1 and that the extension remains within the timing/size constraints of existing standards. Further, as the proposed extension is compatible with the roles and data flows of existing EV charging standards, we argue that requirement FR_2 (minor changes) is met. Moreover, as our extension is designed to fully support existing features (semi-offline operation, CP reservation, certificate pools, etc.), FR_3 is arguably met.

6 Formal Verification of Security and Privacy

We analyze the security of our proposed extension in the symbolic model, also called the Dolev-Yao model [12]. In this model, cryptographic primitives are represented by symbolic functions and assumed to be perfectly secure. Thus, the focus lies on the security of the composition of these primitives. Usually, the adversary has complete control over the network (cf. the network hacker adversary type in Section 3).

We use the Tamarin prover [40] to show the security of our model. Tamarin is a state-of-the-art tool for automated symbolic protocol verification.

In Tamarin, a model specifies a set of rules that define the communication and data processing steps of a protocol. First-order logic is used to define trace properties that should hold for all possible execution traces of the model. To verify these properties, Tamarin starts with a state where the property has been violated and performs a backward search over the possible rule executions to determine whether there is a valid path that leads to this state. If there is, the property does not hold and Tamarin has found a counterexample. If there is no possible execution path that can lead to a violation of the property, this proves that the property holds.

Tamarin further supports observational equivalence properties that show that two protocol instances that differ in a term are indistinguishable by the adversary. For example, the adversary cannot distinguish whether an action was taken by actor A or actor B. Hereby, Tamarin computes all possible executions of the rules for adversarial behavior for both models (one where actor A is acting and one where B is acting) and verifies that an equivalent execution exists for the other model.

We make use of trace properties to verify the security requirements of our model and observational equivalence properties for the privacy requirements. Tamarin's support for a mutual global state as well as flexible, user-definable equational theories makes it well suited for our analysis. There are multiple predefined heuristics for automated proof generation as well as support for manual proof guidance and custom heuristics, which are especially important for the rather complex proofs of observational equivalence properties.

In the following, we exemplary introduce *injective agreement* [39], which is the most commonly used notion to prove strong authentication properties in the symbolic model, and its representation as a Tamarin lemma for those who are not familiar with the tool. However, for a better understanding of Tamarin, refer to [52].

Definition 1 (Injective Agreement)**.** *A protocol guarantees* injective agreement *to an honest initiator A with an honest responder B on a set of data items $\vec{ds}$ if, whenever A, acting as initiator, completes a run with the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A, and B was acting as responder in this run. Moreover, each run of A corresponds to a unique run of B and both agents agree on the values of the variables in $\vec{ds}$.*

Listing 1 shows how this notion can be represented in Tamarin's specification language, given that the referenced events (`Commit`, `Running`, `Honest`, `Reveal`) have been defined in the protocol's model accordingly. `Commit(A, B, ds)` (Line 3) means that A completed a run of the protocol, apparently with B, for the data values $\vec{ds}$. Hereby, `@i` denotes that this event occurred at timepoint `i`. For every possible execution trace of the model where such an event occurs, Lines 4-11 have to be satisfied. Lines 4-5 specify the required previous protocol run by B (apparently with the same A and $\vec{ds}$ at timepoint `j` before `i`). Lines 6-8 formulate the required uniqueness of this run, meaning that there cannot be a different completed run (not at the same timepoint `i2` = `i`) of any $A2$ for the same data values $\vec{ds}$.

While there could be multiple corrupted parties in the network, A and B are required to be honest. The lemma is not violated if the adversary can authenticate as a party that is already under her control (Lines 10-11). This is denoted by an `Honest` event for these parties in the specification of the rule with the commit event, i.e., for all `Commit` events between A and B at timepoint i, there is an `Honest` event for A and for B at i generated by the model rules. Thus, the actor C can only be instantiated with A or B for the restriction in Line 11 to hold. The `Reveal` event occurs whenever a party is corrupted and all its secrets are leaked to the adversary.

```
1  lemma injective_agreement:
2  " All A B ds #i.
3      Commit(A, B, ds) @i
4      ==> ( Ex #j. Running(B, A, ds) @j
5          & j < i
6          & not (Ex A2 B2 #i2.
7                 Commit(A2, B2, ds) @i2
8                 & not (#i2 = #i))
9          )
10         | (Ex C #r. Reveal(C) @r
11                & Honest(C) @i)
12 "
```

Listing 1. Injective Agreement Lemma in Tamarin (cf. [52])

Given the specification of the model and the lemmas for the required security properties, Tamarin can verify if the lemmas are satisfied for every possible execution trace of the model by either generating a proof that shows that a path violating the lemma would also contradict the model or by providing a counterexample.

Our formal analysis is based upon the work of Wesmeyer et al. [61], who provide a very fine-grained Tamarin model of TPM-based DAA. We adapt their model to our proposed PnC extension. The changes/extension to the model of [61] can be summarized as follows: (i) Adaptions/additions to the model rules in order to represent the e-mobility-specific roles/processes within the context (e.g., the introduction of the CPS role between the host and issuer in the credential provisioning process or the validation of charge authorization via the eMSP in addition to the DAA signature verification by the CP/verifier), (ii) changes to model rules in order to represent the changes to the base DAA join- and certify processes (cf. Section 4.4 and 4.5; e.g., the possibility of no live interaction with the credential issuer during installation due to the use of credential pools or the binding of session keys to the ISO 15118 session), (iii) definition of new lemmas in order to verify correctness as well as the identified security/privacy requirements (cf. Section 3), and (iv) performance optimizations such that the new models terminate within a reasonable time frame via the adaption of oracles (to work with the new/changed rules) and the addition of source/reuse lemmas. Our changes roughly amount to 54 new rules, 16 changed rules, 32 new lemmas, 18 changed lemmas, and 5 adapted oracles.

As in [61], we use multiple Tamarin models in our analysis due to the complexity of the protocol and proofs. These models and lemmas/proofs are available in our GitLab repository.[6] The repository also includes instructions for running the models (for reproducibility of the formal analysis) and details on verification times. In short, on a standard laptop, the verification times for security-related models range from 14 to 20 minutes and for privacy-related models from 9 to 333 minutes.

For the verification of the security requirements, we assume that the backend communication is done via a secure channel, e.g., via TLS. However, the adversary has complete control over the CP's communication (cf. the local adversary in Section 3). We use separate Tamarin models to verify the credential installation process, which we simplify in the other models, and for online and offline charge authorization to reduce proof complexity. Note that SR_1 is addressed by the usage of a TPM and by conditioning key usage on a `TPM2_PolicyPCR` policy. As discussed in our adversarial model, we consider credentials stored in the TPM as being secure from extraction for our formal analysis. Moreover, we follow the assumption from [7, 61] that the TPM and its host (the EV) communicate via a secure channel. In the following, we list the intuitive descriptions of the formal Tamarin lemmas, which are used to proof the respective security properties.

SR_2 *Secure Credential Installation:* First, we verify that the EV and the CCH (acting as CPS) injectively agree (cf. Definition 1) on credential requests (C_{CReq}), that is, that for all credential requests received by the CCH, apparently from an honest EV V (identified by its platform endorsement key EC_{pk}), V has previously sent a credential request and they agree on its content. In addition, each credential request received by the CCH corresponds to a unique request sent by V. The Tamarin definition of this property is exemplary shown in Listing 2.
Regarding Listing 2, `CommitCPS(CPS, pke, n)` means that the CPS `CPS` accepted a credential request from a vehicle with platform endorsement key `pke` and nonce `n`. `RunningEV(pke, CPS, n)` means that the vehicle identified by `pke` sent a credential request with nonce `n` to `CPS`. `i`, `i2`, `j` and `kr` represent time points at which an event occurred. Injective agreement does not have to hold if one of the involved parties has been corrupted by the adversary (Lines 12 to 14).
Second, regarding the security of credential installation, we show that the EV and the eMSP injectively

6 https://code.fbi.h-da.de/seacop/daa-pnc-tamarin

```
lemma auth_injective_agreement_CPS_EV:
" All CPS pke n #i.
(   ( CommitCPS(CPS, pke, n) @i)
    ==>
    ( ( Ex #j. (RunningEV(pke, CPS, n) @j
        & (#j < #i)
        & ( not(Ex CPS2 pke2 #i2. (
            CommitCPS(CPS2, pke2, n) @i2
            & not(#i2 = #i)
          ) ) )
      ) )
      | (Ex RevealEvent Entity #kr.
            KeyReveal(RevealEvent, Entity)
                @kr
            & Honest(Entity) @i)
    )
) "
```

Listing 2. Tamarin Lemma for Injective Agreement Between the CPS and the EV for Credential Installation (SR_2)

agree on the credentials issued by the eMSP, i.e., all contract credentials, apparently issued by an honest eMSP I that an honest EV V installs have been previously issued by I for V. Moreover, each credential installation corresponds to a unique credential issue event. Third, we verify the secrecy of eMAID keys and installed DAA credentials, i.e., the adversary does not learn the eMAID keys and credentials issued by an honest issuer to an honest EV. We use the syntactic notion of secrecy.

SR_3 *Secure Charge Authorization:* We verify that the eMSP and the EV injectively agree on the charge authorization, that is, for all charge events authorized by an honest eMSP I, apparently initiated by an honest EV V, V previously requested this charge authorization, they agree on its parameters, and each authorization by an eMSP corresponds to a unique charge authorization request by an EV. In addition, we verify that the secrecy of DAA credentials and eMAID keys issued by an honest eMSP to an honest EV is maintained during this process.

SR_4 *Charge Data Authenticity:* Analogous to SR_3, we show that there is an injective agreement between the eMSP and the EV on the billing relevant charge data and its binding to a previous charge authorization. The secrecy properties of DAA credentials and eMAID keys are verified using the same lemma as for SR_3, as SR_3 and SR_4 share the same Tamarin model.

In our privacy analysis, we use the notion of *user-controlled unlinkability* defined for DAA in [6] as it is used by [61]. While there are more general notions of Unlinkability (cf. [2]), this notion was explicitly introduced as a security property for DAA and has been used in previous formal verification of DAA schemes ([61, 62]). Using Tamarin's observational equivalence mode, we show that the adversary cannot decide if two protocol runs were initiated by the same EV. We use separate Tamarin models for each property. Moreover, we assume malicious CPs (cf. the local adversary) and adversarial backend operators (cf. the curious operator) unless they are explicitly allowed linkability based on the respective privacy requirement (cf. Section 3). Malicious CPs and adversarial backend operators are implicitly modeled as part of the adversary's behavior. To simplify the proof, TPM and EV have been collapsed to a single entity in these models. We use online CPs, however, this mainly affects the timing of messages rather than the information the adversary can derive.

Note that while the focus of the privacy evaluation lies on the formally verifiable unlinkability properties, our extension was nonetheless designed to limit the dissemination of personal data to the minimum that is required for the respective actors' operation as discussed in Section 2.3, i.e., the unnecessary distribution of personal data (marked with (X) in Table 1) can be avoided (PR_1).

PR_2 *Unlinkable Credential Installation:* We show that for two honest EVs V_1 and V_2, an honest CCH CPS and an honest eMSP I, the adversary (collaborating with the involved CP and CPO) cannot distinguish between two instances of the protocol, where in the first instance a credential installation process with CPS and I is executed for V_1 and V_2 each, whereas in the second instance, both credential installation processes are executed for V_2. Thus, the adversary cannot link two credential installations as originating from the same EV, even when colluding with the CP and CPO.

PR_3 *Unlinkable Charge Authorization:* Analogously, we show that the adversary cannot distinguish between two charge authorization processes that have been initiated either by the same or by different vehicles. Thus, the adversary cannot link charge authorization messages.

PR_4 *Unlinkable CDRs:* Analogous to PR_2 and PR_3, we show that the adversary cannot distinguish between two charge data attestation processes for either the same or two different vehicles.

PR_5 *Unlinkability of EV Users and Locations:* From PR_3 and PR_4 we know that CPs, CPOs, and the CCH can generally not link a user's sessions (independent of location). For PR_5 we additionally show that an honest-but-curious eMSP I cannot link two charging processes by the same EV V to a location (CP). In detail, V is modeled to charge at two apparently different CPs (CP_1 and CP_2) and I receives the respective authorization and charge data. We verify that I cannot distinguish between two runs of the protocol where V charges either once at CP_1 and CP_2 each or twice at CP_2.

Note that since collusion between the different backend actors is excluded in the adversary model (cf. Section 3), the privacy guarantees do not hold if these actors do collude. Specifically, an honest-but-curious CP/CPO could collude with a CPS in order to link credential installations to each other as well as to link charge sessions (that include an installation) to a location. Additionally, an honest-but-curious CP/CPO could collude with an eMSP in order to link charge authorizations to each other and link CDRs to each other as well as to link charge sessions to a location. Thus, for a real-world use of the solution, the issue of collusion would have to be addressed separately (e.g., via regulations). Notably, the fact that linkability via colluding CPOs and eMSPs is possible may help with the real-world deployability of the solution since it allows for auditability, e.g., in case of a dispute between CPOs and eMSPs. Any such auditability feature would, however, need to be regulated in order to prevent misuse.

7 Related Work

Due to the increasing integration of information and communication technology into vehicles, privacy is becoming more and more important [23]. Pervasive, automatic tracking of vehicles is possible, with the resulting threats to individual privacy [5]. [34] lists privacy issues for electric mobility and [20] privacy challenges in EV charging, including concealed charge data aggregation, charge session unlinkability, identity-/location privacy, and privacy-preserving billing.

Several EV-related privacy issues have been previously addressed. In [32], a blockchain-based approach for privacy-preserving selection of a CP based on tariff options and travel distance is presented. Privacy-preserving matching of EVs with CPs or other EVs for vehicle-to-vehicle charging is discussed in [64]. Privacy-preserving reservation of CPs is discussed in [31, 37]. A privacy-preserving method for EV charge scheduling is presented in [46].

Closer to our work are papers focusing on EV charge authorization and billing. Table 4 compares these papers with our work. In [35, 36], a pseudonymous authentication protocol for EV authentication in a dynamic, contactless charging scenario (i.e., using charging pads that are integrated into the road), supporting the billing of EV users is presented. The protocol, however, requires an out-of-band installation of the EV's credentials and the considered dynamic, contactless charging scenario does not make it ideal for "normal" EV charging (EV charging at a single CP instead of driving over a sequence of charging pads). In [21], an architecture for the automatic and privacy-preserving contract-based charging and billing of EVs via ISO 15118 is presented. It uses anonymous credentials, group signatures, and requires the introduction of a trusted third party. It is not fully compatible with existing roles and data flows and does not consider CP reservations. While, to the best of our knowledge, no formal security analysis exists, [14] conducts a formal privacy analysis of [21] and improvements are suggested in order to address found weaknesses. A privacy-preserving approach for roaming EV charging and billing based on smart cards is discussed in [42]. The reliance on smart cards, however, raises functional issues as the more user-friendly PnC mechanism cannot be supported. Additionally, functional aspects such as offline authorizations and CP reservations are not considered and no formal privacy analysis is conducted.

Closest to our work are [66] and [65]. [66] presents a TPM-based DAA scheme for EV authentication. It supports EV charging and billing including a CP reservation but neither offline charge authorizations nor contract-based charging. Furthermore, it does not consider the integration into existing PnC standards or compatibility with existing roles and data flows. A high-level TPM-based DAA scheme is also proposed in [65]. The authors consider contract-based EV roaming with focus on ISO 15118 PnC. However, they do not consider the entire EV charging architecture and backend protocols in detail and can not (without changes) support important features of current EV charging protocols such as offline operation of CPs, CP reservations, and the automatic installation of EV contract credentials. Additionally, neither [66] nor [65] provide an implementation-based evaluation of their solution's imposed overhead or a formal security-/privacy analysis, thereby lowering the level of confidence in their targeted functional-,

Table 4. Comparison with Related Work

	[35, 36]	[14, 21]	[42]	[66]	[65]	[16–19]	This Work
Privacy							
Protocol Data Minimization Considerations (PR_1):	n/a*	✔	n/a*	n/a*	✔	✘	✔
Unlinkable Automatic Credential Installation (PR_2):	n/a†	n/a†	n/a†	n/a†	n/a†	✘	✔
Unlinkable Charge Authorization (PR_3):	✔	✔	✔	✔	✔	✘	✔
Unlinkable Billing-Relevant Data (PR_4):	✔	✔	✔	✔	✔	✘	✔
Unlinkability of Users and Locations (PR_5):	✔	✔	✔	✔	✔	✘	✔
Formally Verified Privacy Guarantees:	✘	✔	✘	✘	✘	n/a‡	✔
Security							
Hardware-Based Credential Security (SR_1):	✘	✘	✔	✔	✔	✔	✔
Secure Automatic Credential Installation (SR_2):	n/a†	n/a†	n/a†	n/a†	n/a†	✔	✔
Secure Charge Authorization (SR_3):	✔	✔	✔	✔	✔	✔	✔
Secure Billing Data Authenticity (SR_4):	✔	✔	✔	✔	✔	✔	✔
Formally Verified Security Guarantees:	✘	✘	✔	✘	✘	✘	✔
Functional							
Integration in Existing Standards (FR_2):	✘	✔	✘	✘	✔	✔	✔
No Additional Trusted Third Parties (FR_2):	✔	✘	✘	✘	✔	✔	✔
Support for Automatic Credential Installation (FR_3):	OOB	OOB	✘	✘	✘	✔	✔
Support for PnC Authorization (FR_3):	✔	✔	✘	✔	✔	✔	✔
Support for Offline Charge Authorization (FR_3):	✔	✔	✘	✘	✘	✔	✔
Support for CP Reservations (FR_3):	✘	✘	✘	✔	✘	✔	✔
Implementation-Based Overhead Evaluation (FR_1):	✔	✔	✘	✘	✘	✔	✔

✔= Considered in the respective paper(s); ✘= Not considered in the respective paper(s)
n/a = Not applicable to the respective paper(s); OOB = Out-of-Band, i.e., requires separate communication channel
*Not applicable since existing standards/protocols are not considered.
†Not applicable since support for automatic credential installation is not considered or handled via a separate secure channel.
‡Not applicable since no privacy guarantees are considered.

security-, and privacy properties. Using a TPM in EVs, e.g., for securing an EV's PnC credentials, has been discussed in [16–19]. However, privacy is not considered and no formal security analysis is conducted. In contrast to related work, our work addresses the full range of features and processes related to EV charge authorization and billing and we also provide a formal analysis of security and privacy properties.

8 Conclusion

Privacy, especially considering the European General Data Protection Regulation, is important for the EV charging architecture. However, privacy is currently not really considered by related protocols and personal data is unnecessarily revealed to a variety of entities. In this work, we analyzed the current EV charging architecture and identified security, privacy, and functional requirements for integration of privacy-preserving mechanisms. In contrast to related work, we consider the entire PnC EV charging architecture including EV, CP, CPO, CCH, and eMSP as well as important features such as offline operation of CPs, CP reservations, and the automatic installation of EV contract credentials. We propose a TPM-based DAA scheme for privacy-preserving PnC authentication which can be easily integrated into existing protocols with minor changes in message definitions and message handling while not requiring fundamental changes to functions of involved entities or the overall message flow. This maintains compatibility in case intermediate actors do not support our scheme. Our proof-of-concept implementation shows the feasibility of our solution and that it can be integrated into EV charging protocols with only minimal overhead, thus, fulfilling the identified functional requirements. Our formal analysis using Tamarin shows that also the identified security and privacy requirements are fulfilled.

Acknowledgments

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science

and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] M. Abomhara and G. M. Køien. Security and privacy in the internet of things: Current status and open issues. In *2014 international conference on privacy and security in mobile systems (PRISMS)*, pages 1–8. IEEE, 2014.

[2] D. Baelde, S. Delaune, and S. Moreau. A Method for Proving Unlinkability of Stateful Protocols. In *33rd IEEE Computer Security Foundations Symposium*, 33rd IEEE Computer Security Foundations Symposium, CSF 2020, Boston, MA, USA, June 22-26, 2020, Boston, United States, June 2020. URL https://hal.archives-ouvertes.fr/hal-02459984.

[3] R. Barbulescu and S. Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32(4):1298–1336, 2019.

[4] T. Bisztray and N. Gruschka. Privacy impact assessment: Comparing methodologies with a focus on practicality. In A. Askarov, R. R. Hansen, and W. Rafnsson, editors, *Secure IT Systems*, pages 3–19, Cham, 2019. Springer International Publishing. ISBN 978-3-030-35055-0.

[5] M. Bradbury, P. Taylor, U. I. Atmaca, C. Maple, and N. Griffiths. Privacy challenges with protecting live vehicular location context. *IEEE Access*, 8:207465–207484, 2020.

[6] E. Brickell, L. Chen, and J. Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *Int. J. Inf. Secur.*, 8(5):315–330, sep 2009. ISSN 1615-5262. 10.1007/s10207-009-0076-3. URL https://doi.org/10.1007/s10207-009-0076-3.

[7] J. Camenisch, L. Chen, M. Drijvers, A. Lehmann, D. Novick, and R. Urian. One tpm to bind them all: Fixing tpm 2.0 for provably secure anonymous attestation. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 901–920. IEEE, 2017.

[8] Chaos Computer Club. Chaos computer club hacks e-motor charging stations. https://www.ccc.de/en/updates/2017/e-motor, 2017. Last visited on 26/07/2021.

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and T. Polk. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280, RFC Editor, 5 2008. URL http://www.rfc-editor.org/rfc/rfc5280.txt.

[10] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.

[11] Q. Do, B. Martini, and K.-K. R. Choo. The role of the adversary model in applied security research. *Computers & Security*, 81:156–181, 2019.

[12] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2): 198–208, 1983.

[13] ElaadNL. EV related protocol study, 1 2017. URL https://www.elaad.nl/research/ev-related-protocol-study/. Arnhem, The Netherlands.

[14] M. Fazouane, H. Kopp, R. W. van der Heijden, D. Le Métayer, and F. Kargl. Formal verification of privacy properties in electric vehicle charging. In *International Symposium on Engineering Secure Software and Systems*, pages 17–33. Springer, 2015.

[15] G. Friedland and R. Sommer. Cybercasing the joint: On the privacy implications of geo-tagging. In *HotSec*, pages 1–6, 2010.

[16] A. Fuchs, D. Kern, C. Krauß, and M. Zhdanova. TrustEV: Trustworthy electric vehicle charging and billing. In *Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing SAC 2020*. ACM, 2020. https://doi.org/10.1145/3341105.3373879.

[17] A. Fuchs, D. Kern, C. Krauß, and M. Zhdanova. Securing electric vehicle charging systems through component binding. In *39th International Conference on Computer Safety, Reliability and Security, SAFECOMP*. Springer, September 2020. https://doi.org/10.1007/978-3-030-54549-9_26.

[18] A. Fuchs, D. Kern, C. Krauß, and M. Zhdanova. HIP: Hsm-based identities for plug-and-charge. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388337. 10.1145/3407023.3407066. URL https://doi.org/10.1145/3407023.3407066.

[19] A. Fuchs, D. Kern, C. Krauß, M. Zhdanova, and R. Heddergott. HIP-20: Integration of vehicle-hsm-generated credentials into plug-and-charge infrastructure. In *Computer Science in Cars Symposium*, CSCS '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376211. 10.1145/3385958.3430483. URL https://doi.org/10.1145/3385958.3430483.

[20] W. Han and Y. Xiao. Privacy preservation for v2g networks in smart grid: A survey. *Computer Communications*, 91: 17–28, 2016.

[21] C. Höfer, J. Petit, R. Schmidt, and F. Kargl. Popcorn: privacy-preserving charging for emobility. In *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*, pages 37–48, 2013.

[22] M. Howard and S. Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.

[23] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy*, 2(3): 49–55, 2004.

[24] A. Humayed, J. Lin, F. Li, and B. Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4 (6):1802–1831, 2017.

[25] S. Hussain, A. Kamal, S. Ahmad, G. Rasool, and S. Iqbal. Threat modelling methodologies: a survey. *Sci. Int.(Lahore)*, 26(4):1607–1609, 2014.

[26] Infineon. A safe for sensitive data in the car: Volkswagen relies on tpm from infineon, 1 2019. URL https://www.infineon.com/cms/en/about-infineon/press/market-news/2019/INFATV201901-030.html.

[27] ISO/IEC. Road vehicles – vehicle to grid communication interface – part 1: General information and use-case definition. ISO Standard 15118-1:2013, International Organization for Standardization, Geneva, Switzerland, 4 2013.

[28] ISO/IEC. Road vehicles – vehicle-to-grid communication interface – part 2: Network and application protocol require-

ments. ISO Standard 15118-2:2014, International Organization for Standardization, Geneva, Switzerland, 4 2014.

[29] ISO/IEC. Road vehicles – vehicle to grid communication interface – part 20: Network and application protocol requirements. ISO/FDIS 15118-20, International Organization for Standardization, Geneva, Switzerland, 08 2021.

[30] D. Kern. Privacy-preserving architecture for ev charging and billing. Master's thesis, TU Darmstadt, 2021.

[31] V. T. Kilari, R. Yu, S. Misra, and G. Xue. Robust revocable anonymous authentication for vehicle to grid communications. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4845–4857, 2020.

[32] F. Knirsch, A. Unterweger, and D. Engel. Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. *Computer Science-Research and Development*, 33(1-2):71–79, 2018.

[33] G. M. Køien. A privacy enhanced device access protocol for an iot context. *Security and Communication Networks*, 9(5): 440–450, 2016.

[34] L. Langer, F. Skopik, G. Kienesberger, and Q. Li. Privacy issues of smart e-mobility. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pages 6682–6687. IEEE, 2013.

[35] H. Li, G. Dan, and K. Nahrstedt. Portunes: Privacy-preserving fast authentication for dynamic electric vehicle charging. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 920–925. IEEE, 2014.

[36] H. Li, G. Dán, and K. Nahrstedt. Portunes+: Privacy-preserving fast authentication for dynamic electric vehicle charging. *IEEE Transactions on Smart Grid*, 8(5):2305–2313, 2016.

[37] J. K. Liu, W. Susilo, T. H. Yuen, M. H. Au, J. Fang, Z. L. Jiang, and J. Zhou. Efficient privacy-preserving charging station reservation system for electric vehicles. *The Computer Journal*, 59(7):1040–1053, 2016.

[38] J. Lopez, R. Rios, F. Bao, and G. Wang. Evolving privacy: From sensors to the internet of things. *Future Generation Computer Systems*, 75:46–57, 2017.

[39] G. Lowe. A hierarchy of authentication specifications. In *Proceedings 10th Computer Security Foundations Workshop*, pages 31–43. IEEE, 1997.

[40] S. Meier, B. Schmidt, C. Cremers, and D. Basin. The tamarin prover for the symbolic analysis of security protocols. In *International Conference on Computer Aided Verification*, pages 696–701. Springer, 2013.

[41] J.-P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel. Attacker model for connected and automated vehicles. In *ACM COMPUTER SCIENCE IN CARS SYMPOSIUM*, 2018.

[42] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan. Roaming electric vehicle charging and billing: An anonymous multi-user protocol. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 939–945. IEEE, 2014.

[43] OCA. Open Charge Point Protocol 2.0.1 - Part 2 - Specification. Open standard, Open Charge Alliance, Arnhem, Netherlands, 3 2020. URL https://www.openchargealliance.org/protocols/ocpp-201/.

[44] A. Paverd, A. Martin, and I. Brown. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. *Tech. Rep.*, 2014.

[45] J. Petit, M. Feiri, and F. Kargl. Revisiting attacker model for smart vehicles. In *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, pages 1–5. IEEE, 2014.

[46] C. Rottondi, S. Fontana, and G. Verticale. Enabling privacy in vehicle-to-grid interactions for battery recharging. *Energies*, 7(5):2780–2798, 2014.

[47] Y. Sakemi, T. Kobayashi, T. Saito, and R. Wahby. Pairing-friendly curves. Internet-Draft draft-irtf-cfrg-pairing-friendly-curves-09, IETF Secretariat, 11 2020. URL http://www.ietf.org/internet-drafts/draft-irtf-cfrg-pairing-friendly-curves-09.txt.

[48] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody. Threat modeling: a summary of available methods. Technical report, Carnegie Mellon University Software Engineering Institute Pittsburgh United States, 2018.

[49] L. Sion, K. Wuyts, K. Yskout, D. Van Landuyt, and W. Joosen. Interaction-based privacy threat elicitation. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 79–86. IEEE, 2018.

[50] J. Smart and S. Schey. Battery electric vehicle driving and charging behavior observed early in the ev project. *SAE International Journal of Alternative Powertrains*, 1(1):27–33, 2012.

[51] The Mobility House. Python implementation of the Open Charge Point Protocol (OCPP), 2020. URL https://github.com/mobilityhouse/ocpp. (Accessed 2020-11-07).

[52] The Tamarin Team. *Tamarin-Prover Manual - Security Protocol Analysis in the Symbolic Model*, 2019.

[53] tpm2-software. Source code implementing the Trusted Computing Group's (TCG) TPM2 Software Stack (TSS), 2020. URL https://github.com/tpm2-software/tpm2-tss. (Accessed 2020-12-14).

[54] Trusted Computing Group. Trusted Platform Module Library - Part 1: Architecture. Specification Family 2.0 - Rev. 01.38, 9 2016.

[55] Trusted Computing Group. Trusted Platform Module Library - Part 2: Structures. Specification Family 2.0 - Rev. 01.38, 9 2016.

[56] Trusted Computing Group. TCG EK Credential Profile. Specification Ver. 2.3 - Rev. 2.0, 7 2020.

[57] V2G Clarity. Reference Implementation Supporting the Evolution of the Vehicle-2-Grid communication interface (RISE V2G), 2020. URL https://github.com/V2GClarity/RISE-V2G. (Accessed 2020-11-02).

[58] M. van der Kam and R. Bekkers. Comparative analysis of standardized protocols for ev roaming. Report d6.1 for the evroaming4eu project, Netherlands Knowledge Platform for Public Charging Infrastructure (NKL), 5 2020.

[59] VDE. Handling of certificates for electric vehicles, charging infrastructure and backend systems within the framework of iso 15118. VDE-AR-E 2802-100-1:2019-12, 12 2019.

[60] S. Wesemeyer, C. J. Newton, H. Treharne, L. Chen, R. Sasse, and J. Whitefield. Source code for "Formal Analysis and Implementation of a TPM 2.0-based Direct Anonymous Attestation Scheme" accepted at ASIACCS 2020, 2020. URL https://github.com/UoS-SCCS/ecc-daa. (Ac-

cessed 2020-12-14).

[61] S. Wesemeyer, C. J. Newton, H. Treharne, L. Chen, R. Sasse, and J. Whitefield. Formal analysis and implementation of a tpm 2.0-based direct anonymous attestation scheme. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 784–798, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367509. 10.1145/3320269.3372197. URL https://doi.org/10.1145/3320269.3372197.

[62] J. Whitefield, L. Chen, R. Sasse, S. Schneider, H. Treharne, and S. Wesemeyer. A symbolic analysis of ecc-based direct anonymous attestation. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 127–141. IEEE, 2019.

[63] K. Wuyts, D. Van Landuyt, A. Hovsepyan, and W. Joosen. Effective and efficient privacy threat modeling through domain refinements. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1175–1178, 2018.

[64] F. Yucel, K. Akkaya, and E. Bulut. Efficient and privacy preserving supplier matching for electric vehicle charging. *Ad Hoc Networks*, 90:101730, 2019.

[65] D. Zelle, M. Springer, M. Zhdanova, and C. Krauß. Anonymous charging and billing of electric vehicles. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–10, 2018.

[66] T. Zhao, C. Zhang, L. Wei, and Y. Zhang. A secure and privacy-preserving payment system for electric vehicles. In *2015 IEEE International Conference on Communications (ICC)*, pages 7280–7285. IEEE, 2015.