# Self-sovereign Identity for Electric Vehicle Charging

Adrian Kailus[1], Dustin Kern[2(✉)], and Christoph Krauß[2]

[1] DB Systel GmbH, Frankfurt, Germany
a@kailus.dev
[2] Darmstadt University of Applied Sciences, Darmstadt, Germany
{dustin.kern,christoph.krauss}@h-da.de

**Abstract.** Electric Vehicles (EVs) are more and more charged at public Charge Points (CPs) using Plug-and-Charge (PnC) protocols such as the ISO 15118 standard which eliminates user interaction for authentication and authorization. Currently, this requires a rather complex Public Key Infrastructure (PKI) and enables driver tracking via the included unique identifiers. In this paper, we propose an approach for using Self-Sovereign Identities (SSIs) as trusted credentials for EV charging authentication and authorization which overcomes the privacy problems and the issues of a complex centralized PKI. Our implementation shows the feasibility of our approach with ISO 15118, meaning that existing roles/features can be supported and that existing timing/size constraints of the ISO standard can be met. The security and privacy of the proposed approach is shown in a formal analysis using the Tamarin prover.

**Keywords:** Electric Vehicle · Privacy · Plug and Charge · Self-Sovereign Identity · ISO 15118

## 1 Introduction

Plug-and-Charge (PnC), e.g., using the standard ISO 15118, enables Electric Vehicles (EVs) to charge without user interaction at public Charge Points (CPs) operated by a Charge Point Operator (CPO). The EV stores relevant data such as contract credentials and automatically performs all necessary steps to start a charging session, e.g., authentication, authorization, and negotiation of charging parameters. No RFID cards or smartphone apps are required anymore. To enable this, ISO 15118 defines a complex Public Key Infrastructure (PKI) and uses a unique identifier to identify the user or actually the user's personal charging contract. The charging contract is the basis for billing of PnC sessions and is concluded between an EV user and an e-Mobility Service Provider (eMSP).

The complex PKI architecture of ISO 15118 requires all entities to operate central (sub-) Certificate Authorities (CAs). These entities include CPOs and eMSPs but also Original Equipment Manufacturers (OEMs) and a Contract Clearing House (CCH). OEMs produce EVs and the CCH enables roaming


C. Pöpper and L. Batina (Eds.): ACNS 2024, LNCS 14585, pp. 137–162, 2024.
https://doi.org/10.1007/978-3-031-54776-8_6

services for charging at CPs from different operators. Furthermore, the Root CAs are possible single points of failure. The unique identifier of the charging contract, called e-Mobility Account Identifier (eMAID), enables user tracking which raises privacy issues. By analyzing movement profiles, user habits or even the health status may be deduced, e.g., if the vehicle is regularly charged at a hospital.

To overcome the issues of centralized systems such as PKIs or identity providers, Self-Sovereign Identities (SSIs) gained a lot of attention in the last years. SSI provides a digital identity and enables users to control the information they disclose to prove their identity and to protect their privacy.

In this paper, we propose an approach for using SSIs as trusted credentials for EV charging authentication and authorization. Our approach solves the issues of complex centralized PKI and protects against linking multiple authentication processes. The contributions of this paper are as follows: (i) Concept for the secure integration of SSI into ISO 15118 with privacy-preserving charging authentication/authorization. (ii) Proof-of-concept implementation showing minor additional overhead and easy integration into existing systems. (iii) Formal security and privacy analysis in the symbolic model using the Tamarin prover [35]. (iv) Publishing the used Tamarin models (cf. Sect. 7.2) for reproducibility of the automated proofs and reusability of used modeling concepts in related work.

The remainder of the paper is structured as follows: Sect. 2 describes necessary background to understand our approach. Related work is discussed in Sect. 3. In Sect. 4, we present identified requirements for our concept which is introduced in Sect. 5. Our prototypical implementation is described in Sect. 6, followed by the security, privacy, and practical evaluations in Sect. 7. Finally, we conclude the paper and discuss future work in Sect. 8.

## 2 Background

In this section, we describe background on e-mobility and SSI. The focus is on the certificate-based authentication which we replace with SSI credentials.

### 2.1 E-mobility

Figure 1 shows a simplified e-mobility architecture for AC and DC charging according to the ISO 15118 standard. There exist two editions of the standard, the first edition ISO 15118-2 [21] and the second edition ISO 15118-20 [22] which brings some security improvements. Our solution can be applied to both versions. Other methods for charge control/authentication are out-of-scope, e.g., basic Pulse-Width Modulation (PWM) signaling based on IEC 61851-1 [20] for AC charging, high-level communication via DIN 70121 [4] (which can be seen as a simpler/early version of ISO 15118-2 that only supports DC charging and does not include PnC authorization), or charge authorization via Autocharge [39] (i.e., insecure authorization via the vehicle's MAC address).

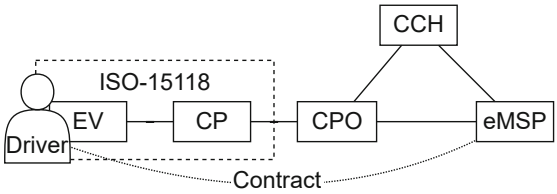


**Fig. 1.** Architecture Overview (cf. [6])

An OEM manufactures the EV (not shown), provides some initial credentials to the EV, and sells it to the new owner. The owner concludes a contract with an eMSP for charging at public CPs which are operated by a CPO. The initial credential from the OEM are used by the EV to request contract credentials from the eMSP. A Certificate Provisioning Service (CPS) establishes trust in the contract credentials provided by the eMSP. The EV stores and uses the contract credentials for PnC authorization and billing during a charging session with a CP. The communication between EV and CP is secured with TLS. The first edition of ISO 15118 uses unilateral TLS authentication of the CP and challenge-response-based authentication of the EV inside the TLS channel. The second edition uses mutual authentication with a vehicle certificate installed by the OEM in addition to the challenge-response-based EV authentication.

ISO 15118 requires multiple certificates and defines a rather complex PKI. The PKI consists of four[1] parts for CPO, OEM, eMSP, and CPS. All PKIs consist of up to two sub-CAs below a root CA. The root CA for CPO- and CPS-PKI is the V2G root CA which usually also certifies the sub-CAs of eMSP and OEM via cross-signing. The eMSP-PKI and OEM-PKI are always certified by their own root CAs.

The CPO-PKI is used for issuing certificates for CPs which are used for CP authentication in the TLS handshake.

The OEM-PKI is used to issue the OEM provisioning certificate which includes the unique identifier Provisioning Certificate Identifier (PCID). The OEM provisioning certificate is used as initial trust anchor for installing the contract credentials. In case the second edition ISO 15118-20 is used, additionally, a vehicle certificate is issued for EV authentication in the TLS handshake.

The eMSP-PKI is used to generate the contract certificate after concluding a contract with an EV owner. The eMSP generates contract certificate data which consists of a private key and the contract certificate (including the corresponding public key, a unique identifier called eMAID, and additional information). This data is installed when the EV is first connected to a public CP. The private key is encrypted with the public key of the OEM provisioning certificate to ensure that only the specific EV can access this key.

Finally, the CPS-PKI is used for generating certificates which are used by a CPS to sign contract certificate data generated by the eMSP. An EV can verify

[1] We omit the part for private environments since it is not relevant for our work.

the signature and the certificate chain up to the known V2G root CA. Thus, the verifier does not need to know the eMSP root CA.

The custom ISO 15118 PKI (with its required trust relations and certificate handling etc.) incurs a high level of complexity, which resulted in critique by relevant stake holders [2] and an importance for complexity-reducing measures [7]. Additionally, as backend communication is out-of-scope for ISO 15118, the PKI definition leaves many open issues such as certificate revocation handling, contract validation, or the handling of contract certificate requests/responses. Addressing these open issues requires proprietary solutions or additional standardization (e.g., the German VDE guideline for ISO 15118 certificate handling [46]), which further increases complexity.

In addition to the complexity of the PKI, there is another issue in ISO 15118 namely the lack of privacy protection. Currently, a lot of information, arguably not required for operation, is disclosed to entities such as CPOs, CCHs, and eMSPs [27]. For example, it would not be necessary to send the exact time and CP location of a charging session to the eMSP or the eMAID to the CPO.

### 2.2 Self-Sovereign Identity (SSI)

A Self-Sovereign Identity (SSI) allows a user to create and fully control a digital identity without requiring centralized infrastructures or identity providers. The user can also control how personal data is shared and used by another party via a decentralized path. After an information is verified by an issuer (e.g., a university verifying a degree), a verifier (e.g., a company) can always trust that information to be true. Subsequently, the information holder (e.g., a student) does not need to provide the full information to the verifier to prove its identity. This is achieved using verifiable credentials (standardized by the W3C [44]), the distributed identity protocol, and a distributed ledger technology (which is mostly a blockchain). The information holder registers an information identifier at a ledger, which is verified by an issuer, and the verifier can trust this information. In the following, we introduce the most relevant terms for our work.

*Verifiable Claims.* In SSI, the essence is that a counterpart can rely on a claim without having control over the content of the claim. Here, a distinction must be made between a Claim and a Verifiable Claim. First, a claim is simply a statement about a fact that anyone could make and without being verifiable. For example, it could be stated that Alice is a graduate of a certain university. However, for this statement to become a Verifiable Claim, the signature of an issuer may be added to it. Alternatively, zero-knowledge cryptography may be used in a privacy-preserving manner to indirectly prove that a claim is covered by a valid verifiable credential [44].

*Verifiable Credentials.* A collection of claims together with an identifier and metadata such as the issuer, expiration date, terms of use, and keys form a credential. Credentials are comparable to conventional ID documents, which

likewise bundle a number of statements. Multiple credentials can be combined into one profile.[2]

*Decentralized Identifiers.* Identifiers that can be resolved to a Distributed Identifier (DID) Document[3] and do not require a central registration authority to be created. The DID Document, which can only be modified by the DID Controller, can contain information about public keys, verification methods, the controller, and authentication methods, among other things. The DID Controller also defines the subject of the DID, e.g., a person or organization. Specific sections in a DID document can be referenced by the respective DID URL. Both the DID and the DID Document are stored in a *Verifiable Data Registry* (e.g., a distributed ledger) and their combination is called a DID Record. The public keys of a DID enable encrypted communication with the owner of the DID. To do this, a communication partner can either use a DID Record they got from the other party or look up the public keys in the *Verifiable Data Registry* [41].

*DID Auth.* There are 10 different architectures to authenticate an identity holder using different transports for the challenge-response cycle [43]. The main focus is to let an identity holder prove to have control over a DID. Authentication can be unilateral or bilateral, with both parties demonstrating control over their own DID. This may also involve the exchange of Verifiable Credentials if required by the use case. There are three ways to combine DID Auth with Verifiable Credentials: DID Auth and the Verifiable Credentials are exchanged separately (in that order); The Verifiable Credentials are part of DID Auth and represent an optional field in the authentication protocol or finally, DID Auth can be considered a special case of a Verifiable Credential, with a claim "I am me". The authentication process is based on a challenge-response cycle where the relying party authenticates the identity holder using, for example, a cryptographic signature.

## 3 Related Work

The increasing integration of information and communication technology into vehicles enables automated tracking of vehicles which threatens the privacy of drivers and passengers [1]. [30] discusses privacy issues for electric mobility and [17] privacy challenges for EV charging.

Several approaches for security and/or privacy in EV charging have been proposed. In [31,32], an EV authentication protocol for contactless charging (i.e., using charging pads integrated into the road) using pseudonyms is proposed. An architecture for privacy-preserving contract-based charging and billing of EVs using ISO 15118 is presented in [19]. A formal analysis and improvements of this architecture are presented in [9]. A privacy-preserving solution for roaming EV charging and billing based on smart cards is proposed in [37]. The solutions

[2] Combining credentials, 2018, https://github.com/w3c/vc-data-model/issues/112.
[3] DID resolution, W3C, 2021, https://w3c-ccg.github.io/did-resolution/.

presented in [27,49,50] all require a Trusted Platform Module (TPM) to realize a Direct Anonymous Attestation (DAA) scheme for EV authentication. In [26], an approach for quantum-secure EV charging is presented. Using a TPM for protecting credentials but without privacy protection is proposed in [13–16]. All approaches still require a complex PKI.

Some work exists that seeks to address the issues around privacy and user profiling when charging EVs via the implementation of a new, anonymous payment channel. This often involves a blockchain solution that promises anonymous payment processing and a decentralized infrastructure. The authors of [10], for example, present a solution where payment for charging electricity is handled through multiple blockchains. A main blockchain negotiates transactions between the operator and the CPs, and on sub-blockchains, multiple customers join together to form credit sharing groups in which individual payments cannot be linked to the buyer of the credits. Here, the degree of anonymity is measured using K-anonymity, which quantifies the group size from which a user is indistinguishable. The main blockchain is connected to the sub-blockchains via a *bridge* role that communicates with credit buyers. The authors of [48] also present a blockchain-based solution for charging EVs, which is also based on *K-Anonymity*. Their approach uses a distributed PKI that separates user registration and verification across two blockchains. Payment here is handled via smart contracts. In [29], a blockchain-based approach for privacy-preserving selection of a CP based on tariff options and travel distance is presented. The authors of [47] propose the implementation of a blockchain-based PKI for Internet of Things (IoT) and demonstrate the feasibility and efficiency of such an IoT PKI through a prototype implementation and experiments. The PKI network is based on Emercoin 15 and uses a proof-of-stake consensus algorithm.

Some work already considers the use of SSI for EV charging. The authors of [42] provide a high-level analysis of the potential benefits that an SSI solution can bring to EV charging. However, no detailed concept is proposed and details on, e.g., the integration into existing EV charging processes or the resulting overhead are not analyzed. Similar to our work is the approach of [18], which also uses SSI for decentralized eRoaming. However, this concept differs from our ISO 15118 extension and makes use of the user's smartphone instead of allowing for PnC-based EV authentication without user interaction. Also, no implementation is developed and a detailed analysis of performance overhead and security is provided.

In contrast to related work, our work presents a novel solution for the integration of SSI into the EV charging ecosystem. We consider the integration into existing protocols and process to enhance the potential usability of the solution as much as possible. Additionally, we provide a performance analysis based on a proof-of-concept implementation as well as a formal security and privacy analysis using the Tamarin prover [35].

## 4 System Model and Requirement Analysis

The following section outlines the scope of this work, defines an attacker model, and discusses the concept requirements, which are grouped into three categories: Functional Requirements (FR), Security Requirements (SR), and Privacy Requirements (PR). We derive our requirements under consideration of the state of the art (cf. Sect. 2) in combination with the attacker model (cf. Sect. 4.2) and considering relevant threat/requirement analyses from related work (cf. Sect. 3).

### 4.1 Scope

Among other things, the PnC process maps a bidirectional authentication between CP and EV to trust the existence of a contractual relationship and to rule out malicious actors. These authentications in ISO 15118 are based on a common PKI, which is used, among other tasks, to authorize a vehicle for a charging process, to authenticate the charging infrastructure, or to establish the TLS connection. In an all-encompassing extension of traditional authentication via PKI and its certificates, both authentications would therefore be replaced, including their use for the TLS connection and the *metering* messages during the charging loop. The scope of this work, however, is limited to the proprietary application-layer-based authentication process of the contract information provided by the EV to the CP. Since the CP's authentication towards the EV uses generic TLS-based methods and has (based on our analysis) no special privacy requirements, we argue that the CP's authentication could be replaced with generic SSI-based methods (cf. *DID Auth* in Sect. 2.2) in a straight-forward manner. This would also address the CP-related PKI requirements. Thus, we do not consider further details of the CP's authentication in this work and instead focus on the EV's side.

### 4.2 Attacker Model

A successful attack in the EV charging context could lead to financial damages, cause safety issues or privacy violations, and may (if large-scale enough) even cause power grid stability issues [8,24,25]. Thus, in order to make the concept viable against possible attacks and vulnerabilities, an attacker model is set up in the following.

Classic attacker models, such as the Dolev-Yao Model [5], outline malicious network participants capable of intercepting network communications, sending and modifying messages. However, we assume that basic cryptographic primitives and implementations hold [36].

Additionally, we consider threats to the system's privacy. The centralized approach to certificate validation makes users traceable and their personal data vulnerable to attack by any of the actors. This threat is increased in case one of the actors is compromised by an attacker or stops following the agreed protocol to obtain additional information. While such *malicious operators* pose a major threat, the danger posed by such *malicious operators* is limited [40]. This is

mainly due to the fact that operators have to comply with legal regulations and maintain their image to the public. Taking this into account, the *Honest-but-Curious Operator* is described below (cf. [27]).

Above all, the *Honest-but-Curious Operator* does not want to create a malicious impression to the outside world by deviating from the agreed protocols. Since involved in the process, such operators use all information available to them to ultimately derive additional benefit from it. In the PnC context, potentially *Honest-but-Curious Operators* can include the CPOs, eMSPs and the CCH. At this point it is assumed that several operators do not accumulate their available information to draw a more comprehensive data picture, since this is opposed to the competition relationship among operators and should additionally be prevented by regulations. Ultimately, the regulation of operators is beyond the control of this concept.

### 4.3 Functional Requirements

In order to ensure user-friendliness and to allow for an easy integration of the solution into existing protocols and processes, we define several functional requirements. The requirements ensure that features of the original ISO 15118 can be supported by the new concept. For example, in order for the vehicle to authenticate itself at the charging stations with its contract information, a process must be defined for contract installations which provide the vehicle with the necessary information. In order to uniquely associate a driver's contract with the vehicle, the vehicle must be uniquely identifiable during the installation process. In order to ensure that the solution is user-friendly, any additional overhead should remain acceptable. Functional Requirements (FR) are listed in the following:

- FR1 Vehicle charging as well as contract installation should still be possible without further user interaction, since this is the concept of PnC.
- FR2 Contract authentication via SSI should be negotiable as an option to the existing authentication methods.
- FR3 All SSI roles should be able to be taken by an actor from the ISO 15118 ecosystem. In SSI, the credential verification process principally covers three roles: the Issuer, the Holder and the Verifier, which must be uniquely applied to an entity in the PnC context for each authentication.
- FR4 The vehicle should continue to manage the necessary authentication information itself (in a wallet).
- FR5 All contract issues from all issuers should fit an agreed schema baseline.
- FR6 As in ISO-15118, it should be possible to delay the installation of the contract information until the first charging process.
- FR7 The charging station should relay communication from the vehicle to the other actors in case the vehicle cannot use cellular.
- FR8 The additional computational- and communication overhead of a SSI-based solution should be minor.

### 4.4 Security and Privacy Requirements

The non-functional requirements for the concept are listed and explained below. This includes Security Requirements (SR) and Privacy Requirements (PR). The security requirements focus on providing secure authentication for the actors involved in relevant processes (setup, credential installation, charging, billing):

- SR1 The setup proceeds of the solution should be secure (e.g., the setup of EVs with provisioning credentials or the setup of eMSPs as issuers of verifiable credentials). That is, all relevant parties should be securely authenticated to enable trust between the parties.
- SR2 During the contract credential installation the eMSP should be able to trust in the originality of the vehicle, similarly to the OEM provisioning certificate in ISO 15118, which is installed during vehicle production. That is, the EV should securely authenticate itself towards the eMSP during the credential installation process.
- SR3 The CP/CPO should be able to trust the EV's provided contract information. That is, the EV should securely authenticate itself towards a CP before the start of a charging process.
- SR4 The contract information should allow the eMSP to associate an invoice from a CPO with a contract. That is, the EV's charge authentication data should securely authenticate the EV's contract towards the eMSP for billing.

The privacy requirements focus non-traceability and non-linkability of EV users:

- PR1 During the authentication process no information should be exchanged that makes the user traceable to either a CPO, CCH or an eMSP, preventing the creation of a user's movement profile (*non-traceability*).
- PR2 A specific CPO, CCH or eMSP should not be able to associate multiple charging operations with individual users (*non-linkability*).

Notably, traceability and linkability of EV users by their eMSP is feasible due to payment processing via traditional payment methods. This problem may be solved by using smart contracts (cf. [48]), which is out-of-scope for this paper.

## 5 SSI Concept

In the following, our concept for integrating an SSI-based solution into the ISO 15118-2 authentication process is developed, including an architectural overview and the message sequences of the communication between the actors. The main challenge is in the specific combination of the different SSI concepts (cf. Sect. 2.2) such that actors, processes, and features of the existing EV charging architecture can still be supported while also designing the concept in a way that enables the (Tamarin-based) symbolic verification of the strong security and privacy requirements (cf. Sect. 4.4). Additionally, we discuss the applicability of the proposed solution to ISO 15118-20.

### 5.1 Concept Overview

In this specific scenario, the already existing parties of ISO 15118 are sufficient to map all three roles *Holder*, *Verifier*, and *Issuer* of the SSI process.

The Holder and the Verifier of the contract authentication process are easy to identify in the PnC context: The Holder is the actor in possession of the contract information. This data could be stored either in a wallet on the driver's smartphone, along with other credentials, or in the EV in the form of an on-board wallet. The first option would require driver consent each time information is accessed from the wallet, similar to [34]. Since the main goal of PnC is to enable vehicle charging without further user interaction, it is preferable to install the wallet in the EV. This also eliminates the need to communicate with the driver's smartphone. Since the verifier needs to authenticate the contracts, this role is taken by the CP, which is already performing this task in ISO 15118.

The issuer first needs access to the original contracts to authenticate them as credentials. This condition applies only to the eMSP, with each eMSP having access solely to the contracts of its clients. Furthermore, the verifiers, i.e., the CPs, should be able to trust the issuer. Since the CPs already had to trust the eMSPs in the conventional ISO 15118, this condition is also met.

To grant multiple issuers write permissions on the Ledger to create documents like *Credential Definitions* or *Credentials*, an additional instance is needed that can give these permissions to the different issuers - the *Steward*.

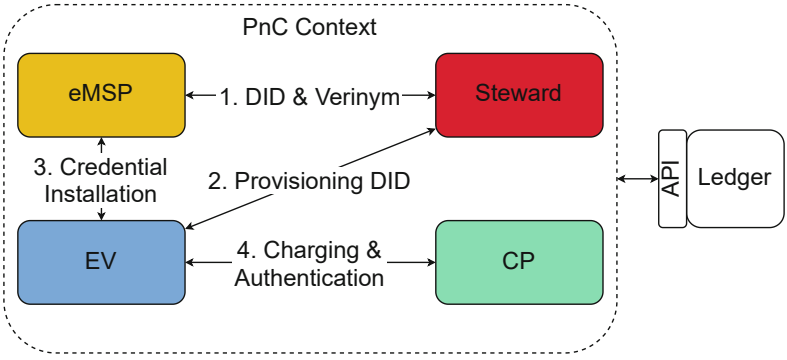


**Fig. 2.** Architecture Overview

Figure 2 shows how these four actors interact for charging authentication in the overall system. Initially, only the steward is authorized to write to the ledger which reduces the number of first-level write permissions. The steward grants second-level write permissions to new eMSPs later on. The steward writes these permissions to the ledger in the form of a verinym (step 1), which enables the eMSP to authenticate its contracts. A verinym is associated with the legal identity of the identity holder [11]. Thus, the legal entity of the eMSP that enters

into the contracts with the customers is associated with the identity on the ledger that has write permissions for the credentials of those same contracts.

In step 2, a *Provisioning DID* is created for the vehicle. This is done before the vehicle is sold. This *Provisioning DID* is necessary to be able to link a specific vehicle to a contract later on. Furthermore, with the help of the public key of a DID, it is always possible for other actors to communicate with its owner in an encrypted way, which will also be helpful later on. Of course, this also applies to all other DIDs used in the PnC context.

Then, in order for the necessary contract information to be authenticated during a charging process, the information must be transferred to the vehicle. This third step can happen once a contract is established and the vehicle has connected to the internet (directly or via a CP). Since the vehicle may have wireless, but this is optional, this step can take place sometime after the *Provisioning DID* has been created between the conclusion of the contract and the charging process. For this, the vehicle requests the credentials from the respective eMSP, which authenticates them on the ledger.

The vehicle can then authenticate itself to the CP during the charging process in the final step 4. Authentication uses *Anoncreds*,[4] i.e., zero-knowledge proofs with Camenisch-Lysyanskaya (CL)-based credentials and paring-based revocation [28]. In short, the EV proofs to the CPs that it possesses valid contract credentials and that these credentials have not been revoked by the issuer (without revealing the actual credentials).

The following sections describe the changes made to the message sequence of ISO 15118 in order to create a working infrastructure for the transition to SSI authentication.

### 5.2 Provisioning DID Creation

Prior to any charging process, the issuer, in this case the eMSP, must be authorized to issue credentials. That is, the eMSP needs write permission to the ledger, which requires publishing its DID (containing a public key) to the ledger. Such a DID is often called a *Verinym*. The eMSP makes a request to the steward, which is authorized to write to the ledger. This process is secured based on prenegotiated secret or public keys. Since both communication partners are legal entities, it can be assumed that there is an agreement between the two in which a secret or public key can be exchanged.

Anther setup process is the creation of a *Provisioning DIDs* (cf. Fig. 3), which is a prerequisite for linking the contract and the vehicle. This process is described in the following paragraphs:

*Step 1.* The EV provisioning process, starts with the production of the vehicle. During this process, the EV creates a *Provisioning DID*, which enables encrypted communication with the EV using the corresponding public key (shared via the ledger; without requiring a traditional PKI). A part of the DID is the *DID*

[4] https://github.com/hyperledger/indy-sdk/tree/main/docs/design/002-anoncreds.

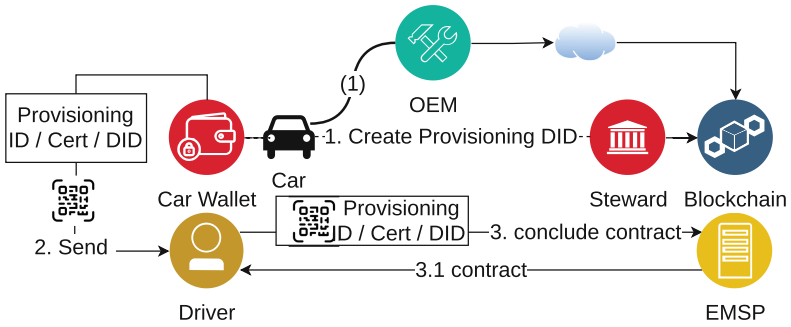


**Fig. 3.** Provisioning *DID* Creation

*record*, which contains the public information for a given DID and must be written to the ledger. In order to write an EV's *Provisioning DID record* on the ledger, communication towards the steward is handled by the EV's OEM on behalf of the EV. After connecting to the steward, the OEM starts with sending an *InitNymReq* with a nonce, answered by the steward with an *InitNymRes*, containing a DID for a key of the steward, the OEM's nonce, a fresh nonce from the steward and the OEM's ID. The *InitNymRes* is signed by the steward (with the key corresponding to the DID) and encrypted with a public key of the OEM. The steward's DID allows the OEM to encrypt future messages to the steward, and the nonces are used to ensure replay-protection and subsequently a proof of possession for the EV's *Provisioning DID*. The OEM creates a *Provisioning DID* (on behalf of the EV and for a provisioning key pair that is provided to the EV), decrypts the steward message, verifies the signature, and signs the steward's nonce with the private key of the *Provisioning DID*.[5] The *Provisioning DID* (including the corresponding public key) and the signature are sent back to the steward, encrypted with the public key from the steward's DID.

*Steps 2 and 3.* When the vehicle is purchased, the *Provisioning DID*, is passed to the user so that the user can pass the *Provisioning DID* to the eMSP and negotiate a contract. The handover at the time of concluding a contract with the eMSP could be via a QR code sent to the user, who then activates the contract by passing on the DID, but other ways are not excluded. Since a potential co-reader does not have the private keys of the DID, he cannot prove their possession and cannot succeed in a challenge. This completes the process until the first charging session.

[5] While it would be possible to generate the *Provisioning DID* key pair in the EV (similar to [13,16]) and have the OEM only collect a signature over the steward's nonce from the EV (which would prevent the EV's private key material from ever leaving the EV), we believe that this method may result in scalability issues. Additionally, one may assume a secure OEM to EV relation during production (in a controlled environment), which limits the security benefit of exclusive key possession by the EV.

### 5.3 Contract Credential Installation

The following is an explanation of the general process steps for installing the Contract Credential (cf. Fig. 4), which requires a *Provisioning DID* and an existing contract with an eMSP. This process is modeled on the Issue Credential Protocol from [12].

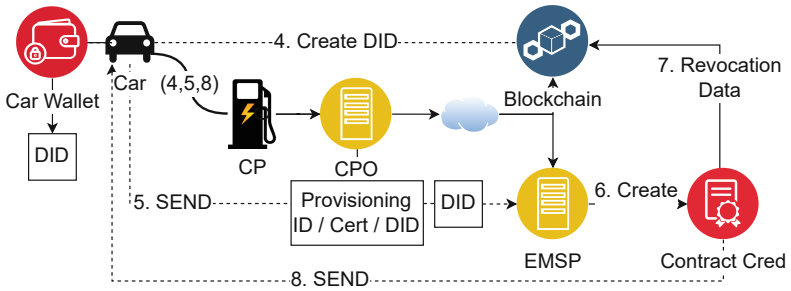


**Fig. 4.** Contract Credential Installation

*Step 4 and 5.* Contract credentials are required in the vehicle during a charging process. To do this, they must first be created and installed in an EV. Similarly to the current installation process in ISO 15118, we tunnel the necessary communication between the EV and the backend via the CP. This method enhances usability as the vehicle may not be able to connect to the Internet, and thus to the ledger and other services, until it is plugged into a CP for the first time. Once the connection is established, the EV starts by sending its *Provisioning DID* to the eMSP. The eMSP responds with its DID and a *Credential Offer*, which includes a nonce and a *Credential Definition ID*. The latter identifies a credential schema, which specifies the structure of all issued credentials (of a certain contract type) by this eMSP with all necessary and optional fields, with public keys, and a *Revocation Registry*. The eMSP's response is encrypted for the EV based on the *Provisioning DID* (i.e., based on the respective *Provisioning DID* public key).

*Steps 6, 7, and 8.* If the EV agrees to this *Credential Offer*, it generates a master secret for the credential. The EV then creates a blinded master secret for the *Credential Offer* and a correctness proof (as per *Anoncreds* definition). Afterwards, the EV builds a *Credential Request* with the blinded master secret and correctness proof and encrypts this request based on the eMSP's DID.

The eMSP decrypts this *Credential Request* and uses it to create the *Contract Credentials* that an EV needs in order to authenticate itself at CPs. Additionally, the eMSP updates the revocation information, i.e., the public tails files and

the accumulator[6] on the ledger to include the new credential. This step can optionally include the revocation of old credentials in case a contract has been terminated or the terms of the contract have changed.

The *Contract Credentials* need to be authenticated by an authorized issuer, which can be the eMSP, and contain all billing-relevant information as attributes. This billing-relevant information, is at least, the eMSP's ID, which is needed by CPs/CPOs to identify the EV user's eMSP for billing purposes. Additionally, the credential attributes can include any tariff information that may be useful to CPs/CPOs (e.g., pricing thresholds or if Vehicle to Grid (V2G) power transfer is supported). The EV user can always decide which attributes from a *Contract Credential* they want to reveal during a zero-knowledge proof.

The EV receives the signed *Contract Credentials* along with the credential revocation information from the eMSP encrypted with the public key of the *Provisioning DID* via the existing connection in a *CreateContractCredentialRes*. The eMSP's response additionally includes a symmetric contract key, which is later used to securely authenticate the EV's contract towards the eMSP for billing purposes. The EV decrypts and verifies the received data and stores it for later authentication during charge sessions.

### 5.4 Charging Process and Credential Validation

The following section will outline the changes to the charging process (cf. Fig. 5). Specifically, the message sequence *Identification, Authentication, and Authorization* from ISO 15118 is considered.

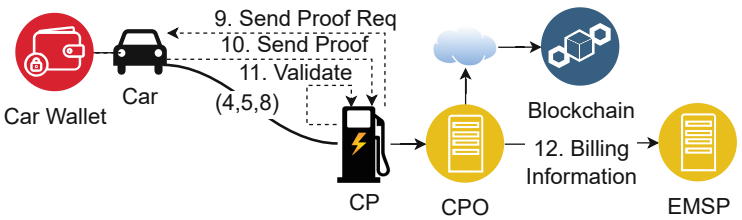


**Fig. 5.** Credential Validation during the Charging Process

*Step 9.* Figure 5 shows the authentication of the vehicle by the CP. In ISO 15118, service parameters such as the payment method are negotiated in the *ServiceDiscoveryReq/-Res*. The authentication method now becomes another service parameter, making *Contract Proof Identification Mode* a third option besides the

[6] https://hyperledger-indy.readthedocs.io/projects/hipe/en/latest/text/0011-cred-revocation/README.html.

existing modes (e.g., PnC). In this *Contract Proof* message sequence, *Identification, Authentication, and Authorization* messages from ISO 15118 are changed after the *PaymentServiceSelectionRes*.

By sending a *RequestProofReq/-Res* the EV receives a proof request from the CP. The CP's proof request includes a nonce and specifies which individual credential attributes the CP expects in its role as verifier, not necessarily all the credentials/attributes issued to the EV by the eMSP.

*Step 10.* From the proof request, the EV then creates a zero-knowledge proof for the requested attributes and a proof of non-revocation using its credential master secret and the CP's nonce. The proofs guarantee to a verifier that the EV possesses valid non-revoked credentials for the identified attributes. The EV additionally uses its symmetric contract key to authenticate its contract towards the eMSP by generating an HMAC over a hash of the CP's *proof request*, a contract identifier, and a timestamp. The hashed *proof request* is used to bind the contract authentication data to the current CP/session, the contract identifier is used by the eMSP to identify the correct contract and symmetric contract key, and the timestamp is used to prevent replays. The contract authentication data is encrypted for the eMSP and sent together with the proofs to the CP in a *ValidateContractProofReq* message.

*Step 11.* The CP can validate the zero-knowledge proof for the credential attributes by using the eMSP's public key and can validate the revocation status of the corresponding credential by using the eMSP's public tails file and the corresponding accumulator value from the ledger. If all verifications are successful, the CP responds with a *ValidateContractProofRes* to the EV. Thereupon, the charging process can continue as described in ISO 15118.

*Step 12.* Finally, the encrypted contract authentication data (along with other billing relevant data, e.g., meter values) is sent from the CP to its CPO, who can forward this data to the corresponding eMSP. The eMSP can decrypt the contract authentication data and identify the correct contract. Hence, the usual billing relations are still possible, i.e., the CPO can bill the eMSP and the eMSP can bill the EV user. However, the CP/CPO can no longer identify the specific EV user and the eMSP can no longer identify the specific charging location.

### 5.5 Integration into ISO 15118-20

While ISO 15118-2 is still the prominent edition of the protocol today, we already consider the integration of our solution into ISO 15118-20, which can be expected to gain increased adoption in the future. We identify several relevant changes to the credential installation and charge authorization processes with ISO 15118-20 as follows: *(i)* updated cipher suites (e.g., from 128 bit AES to 256 bit AES), *(ii)* mutual authentication during the TLS handshake (instead of unilateral CP authentication), and *(iii)* the option to install multiple different contract credentials into one EV (e.g., for different eMSP charging contracts).

We argue that the proposed solution is also applicable for ISO 15118-20 as follows: *(i)* our solution is independent of specific cryptographic algorithms, *(ii)* the generic TLS based authentication (without application-specific requirements unlike the EV's application-layer authentication) can be replaced with generic DID-based approaches (to prevent the need for a conventional PKI) whereby the EV's authentication should again use *Anoncreds* (to not undermine the gained application-layer privacy), and *(iii)* the installation of multiple contract credentials is possible by repeating the process of Sect. 5.3.

## 6 Implementation

To demonstrate the feasibility of the concept, the contract authentication described therein was implemented during the charging process together with all preceding initiation steps such as the creation of the DIDs or the installation of contract credentials. Our implementation is based on the ISO 15118 reference implementation *RISE-V2G* [45]. In order to compare the concept with the actual state of the standard, we compare our implemented methods with the default *RISE-V2G* implementation.

The reference implementation covers all necessary features to establish comparability to the status quo and at the same time serve as a basis for the implementation of the concept. The project Hyperledger Indy[7] provides an implementation for all necessary SSI-operations, thus the Indy SDK[8] was chosen to be integrated into our prototype. The reference implementation was extended by the steward and the eMSP in addition to the existing services EV and CP. They are responsible for the detailed handling of the schemas, credential definitions, and credentials and interact with the other actors. Our prototypical implementation focuses on the message sequence *Identification, Authentication, and Authorization* and the associated communication between EV and the other services as described in the concept. The actual accounting and communication between the secondary actors is not part of the implementation, as this is not in the scope of ISO 15118. Additionally, the eMSP onboarding, its creation of the three data structures *Credential Schema*, *Credential Definition*, and *Revocation Registry* for the credentials of its customers' contracts and installation of *Provisioning DID* are also realized in the implementation.

The concept provides for the eMSP to use the secure channel established by the exchanged DID to create a *WriteVerinymReq*. In the prototype implementation, however, communication is still secured via the old certificate infrastructure, as this has only been extended to include the EV authentication. The CP continues to authenticate itself via certificates.

## 7 Evaluation

In this section, we evaluate the proposed/implemented solution. Specifically, in Sect. 7.1 we discuss the performance results based on our implementation

[7] Hyperledger, 2021, https://www.hyperledger.org.
[8] Indy SDK, 2021, https://github.com/hyperledger/indy-sdk#libindy-wrappers.

from Sect. 6, in Sect. 7.2 we describe our Tamarin-based symbolic security and privacy proofs, and in Sect. 7.3 we discuss how the concept addresses the defined requirements from Sect. 4.

### 7.1 Performance Measurements

Regarding performance, we evaluate the computational- and communication overhead of the proposed solution in comparison to the default ISO 15118 processes as implemented by *RISE-V2G*. Additionally, we verify that the incurred overhead remains within the existing timing and size constraints of the ISO 15118 standard (relevant constraints are the same for ISO 15118-2 and ISO 15118-20). For both types of overhead, the main changes are within the credential installation and charge authorization processes. Details are shown in Table 1.

**Table 1.** Duration and Size of Charging Session Messages for both Implementations

| Message Name | RISE-V2G | | SSI Impl. | |
|---|---|---|---|---|
| | time [ms] | size [byes] | time [ms] | size [byes] |
| **Credential Installation** | | | | |
| CertificateInstallationReq | 296.0 | 811 | - | - |
| CertificateInstallationRes | 32.8 | 3638 | - | - |
| GetCredOfferReq | - | - | 4.0 | 106 |
| GetCredOfferRes | - | - | 44.613 | 6710 |
| CreateContractCredentialReq | - | - | 134.429 | 2185 |
| CreateContractCredentialRes | - | - | 2603.864 | 5961 |
| **Charge Authorization** | | | | |
| PaymentDetailsReq | 649.8 | 1452 | - | - |
| PaymentDetailsRes | 73.6 | 37 | - | - |
| AuthorizationReq | 129.6 | 13 | - | - |
| AuthorizationRes | 7.5 | 15 | - | - |
| RequestProofReq | - | - | 65.3 | 58 |
| RequestProofRes | - | - | 3.6 | 266 |
| ValidateContractProofReq | - | - | 282.302 | 7281 |
| ValidateContractProofRes | - | - | 136.3 | 55 |

The communication overhead of the proposed solution for credential installation messages is 14,962 bytes in total. The default *RISE-V2G* method requires 4,449 bytes for credential installation. Regarding charge authorization, the messages of the proposed solution are 7,660 bytes in total and the messages of the default *RISE-V2G* method are 1,517 bytes. For comparison, based on our measurements, the total communication overhead of a full 1-h default *RISE-V2G*

charge session with a credential installation and a charge status message interval of 10 s is roughly 20,000 bytes. Notably, the only limit on message sizes of the ISO 15118 standard is a result of its 4 byte payload length field and is 4,294,967,295 bytes ([21], Sect. 7.8.3). Hence, we argue, that the increased overhead of the proposed solution is still acceptable.

For computational overhead, all measurements were performed 1000 times[9] and we report the respective average times (always including processing and message transfer). Regarding credential installation, the mean time of the proposed solution was 2786.9 ms compared to 328.8 ms with the default *RISE-V2G* method. Regarding charge authorization, the mean time of the proposed solution was 487.502 ms compared to 860.5 ms with the default *RISE-V2G* method (mostly due to certificate path validations). The results show good performance for the proposed method, especially considering that credential installation is rarely performed (only if a new contract is concluded or old credentials renewed). Notably, ISO 15118 defines relevant timeouts as: 40 s for generating a certificate installation request, 5 s for receiving a certificate installation response, 40 s for requesting a charge authorization, and 2 s for verifying the authorization ([21], Sect. 8.7.2). Hence, the proposed solution can still meet all relevant limits.

### 7.2 Security and Privacy Analysis with Tamarin

We analyze the security of the proposed solution in the symbolic model using the Tamarin prover [35] and the corresponding files are provided online.[10] Tamarin is a state-of-the-art tool for automated security protocol analysis. By default, analysis is performed in the symbolic model, i.e., assuming a Dolev-Yao adversary [5] with full control over the network who cannot break cryptographic primitives without knowing the respective private key (cf. adversary model in Sect. 4.2).

With Tamarin, protocols are specified using a set of *rules*, which define all relevant communication and processing steps of the protocol. Additionally, security requirements are defined as trace properties (lemmas), which need to hold for all possible execution traces of the protocol, i.e., all traces that can be built with the defined rules. Tamarin performs an exhaustive search for a trace that violates the defined requirements. If a trace is found, this trace serves as a counterexample (a specific attack path that violates the requirement). If no trace is found, the security requirement is proven to be satisfied by the defined protocol.

Furthermore, Tamarin enables the verification of observational equivalence properties, which can be used to show that an adversary cannot distinguish between two protocol runs. Observational equivalence is especially useful in order to verify privacy properties, e.g., by proving anonymity in EV charging by showing that an adversary cannot distinguish between two charge authorizations of different EVs.

[9] The measurements were performed on a Lenovo Thinkpad T480 with Intel® Core™ i5-8250U CPU @ 1.60 GHz × 8, 15.5 GiB Ram, running Ubuntu 20.04.3 LTS 64-bit.

[10] https://code.fbi.h-da.de/seacop/SSI-PnC-Tamarin.

**Security Proofs.**
The security requirements from Sect. 4.4 require authentication between different actors over different data. The most commonly used notion to prove strong authentication properties is defined in [33], namely *injective agreement* (preventing spoofing, replay, etc.). This property is defined as follows:

**Definition 1 (Injective Agreement [33]).** *A protocol guarantees to an initiator A* injective agreement *with a responder B on a set of data items ds if, whenever A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A, and B was acting as responder in his run, and the two agents agreed on the data values corresponding to all the variables in ds, and each such run of A corresponds to a* unique *run of B.*

Using our defined Tamarin model, (See footnote 10) we successfully verify the following security properties based on the notion of injective agreement (cf. Definition 1). For this, we assume one steward and the ledger is modeled as a secure storage, where only authorized entities can write but everyone can read. Communication with the ledger is assumed to be a secure channel as specifics of this communication are not part of our concept, but instead standardized by the respective ledger specification. Additionally, we assume that the long-term key of all actors in a specific protocol run are secure since otherwise, attacks are trivially possible (e.g., if an EV's private provisioning key is leaked to an adversary, this adversary can spoof the affected EV towards an eMSP for contract credential installation). However, in order to keep the needed assumptions as weak as possible, other entities of the same types that are not directly involved in the protocol run can be compromised. For normal signatures/encryptions we use the built-in Tamarin functions. The EV zero-knowledge credential proofs are modeled with custom functions, whereby the EV can create a zero-knowledge proof based on the installed credential and its master secret, which the CP can verify with the eMSP's public key and revocation can be verified via a simple request over an accumulator in the ledger. However, zero-knowledge proofs are modeled without specific cryptographic details, since, in the symbolic model, cryptographic functions are anyway assumed to be secure. Besides the injective agreement-based lemmas to proof the desired security properties, our Tamarin files (See footnote 10) also includes lemmas to verify the correctness of the defined model. That is, correctness lemmas are included to verify that the intended processes can be implemented with the defined rules and without adversary intervention in order to prevent the security properties from being trivially met by an incorrect model (e.g., all possible authentications are trivially secure if no authentication is possible at all). In the following, we describe the verified security properties. Note that the following paragraphs only provide intuitive descriptions of the verified properties as the full proofs are automatically generated with the Tamarin tool based on the defined models. The full formal definitions are part of our Tamarin models (provided online (See footnote 10) for reproducibility).

```
1  lemma auth_emsp_steward_verinym:
2  "All Steward S_DID EMSP Verinym_DID #i .
3    CommitStewardVerinym(Steward, S_DID, EMSP, Verinym_DID) @
       i
4    ==> ( Ex #j .
5          RunningEMSPVerinym(EMSP, S_DID, Verinym_DID) @j
6          & (#j<#i)
7          & not( Ex Steward2 EMSP2 S_DID2 #i2 .
8             CommitStewardVerinym(Steward2, S_DID2, EMSP2,
                  Verinym_DID) @ i2
9            & not(#i2=#i) ) )
10       | ( Ex RevealEvent Entity #kr .
11             KeyReveal(RevealEvent, Entity) @ kr
12             & Honest(Entity) @ i)"
```

**Listing 1.1.** Injective Agreement Lemma in Tamarin

*Secure Setup (eMSP to Steward).* Regarding the secure setup (SR1), we verify that an eMSP and a steward (identified by their DID) injectivly agree on the eMSP's verinym DID (and corresponding public key) during the onboarding process. That is, whenever a steward $S$ accepts an DID for writing on the ledger, apparently from an eMSP $E$, $E$ has previously sent this DID to $S$ and both actors agree on the content of the DID. Additionally, each accepted DID by $S$ corresponds to a unique request from $E$. The only allowed exception is, if the long-term key of one of the parties involved in a specific protocol run was leaked.

The Tamarin lemma, which models the *Secure Setup (eMSP to steward)* security property is shown as an example in Listing 1.1. Hereby, lines 2–6 indicate that for every accepted eMSP verinym DID by as steward (identified by S_DID) at time `i`, there exists an event where the same eMSP has sent this verinym DID to the same steward at time `j` and `j` was before `i`. Lines 7–9 models the uniqueness property of the acceptance by the steward, i.e., it says that there cannot exist another protocol run between the same or different actors (steward2 and EMSP2) where the same verinym DID is accepted. Lines 10–12 model the exception, that the security property can be broken if the long-term keys of one of the actors involved in the protocol (i.e., the actor was assumed honest at time `i`; line 12) run was revealed.

*Secure Setup (Cont.)* Regarding the secure setup (SR1), we additionally verify that a steward and an eMSP injectivly agree on the steward's DID public key during the onboarding process. Furthermore, we verify mutual injective agreement between OEM and steward during the onboarding process of an OEM (see the full Tamarin models (See footnote 10) for details).

*Secure Contract Credential Installation.* Regarding the secure credential installation (SR2), we verify that an EV and an eMSP (identified by their DID)

injectivly agree on a contract credential request and response respectively during the installation process (see the full Tamarin models (See footnote 10) for details). The only allowed exceptions are: *(i)* if the long-term key of one of the parties involved in a specific installation protocol run was leaked or *(ii)* if the long-term keys of a previous OEM to steward setup were leaked.

*Secure Charge Authentication and Authorization.* Regarding the secure charge authentication (SR3), we verify that an EV and a CP injectivly agree on an EV's charge request during the authentication process. Additionally, for secure charge authorization/billing (SR4), we verify that an EV and an eMSP injectivly agree on an EV's charge authorization data for the billing process (see the full Tamarin models(See footnote 10) for details). The only allowed exceptions are: *(i)* if the long-term key of one of the parties involved in a specific installation protocol run was leaked or *(ii)* if the long-term keys of a previous OEM to steward setup were leaked or *(iii)* if the long-term keys of a previous credential installation were leaked.

**Privacy Proofs.**
For our privacy analysis, we mainly focus on the verification of symbolic unlinkability properties. Formally, unlinkability is commonly defined as the adversary's inability to distinguish between a scenario in which the same user is involved in multiple protocol runs with a scenario that involves different users per protocol run [3]. This kind of unlinkability definition has been shown as usable for an automated analysis with Tamarin (based on Tamarin's observational equivalence feature) in the EV charging context by [27]. Specifically, we use Tamarin to prove observational equivalence for a scenario with two protocol runs that may be initiated by the same EV or by different EVs. Our models assume *Honest-but-Curious Operators* (cf. adversary model in Sect. 4.2) and we use separate Tamarin models per property for simplicity. The following descriptions provide an intuitive description of the verified properties and full formal definitions can be found as part of the provided Tamarin models. (See footnote 10)

*Non-traceability.* Regarding preventing the creation of a movement profiles (PR1), we verify unlinkability of EVs/users based on their billing relevant data (as received by the backend). Specifically, we show that for two honest EVs $EV_1$ and $EV_2$, the (*Honest-but-Curious*) adversary cannot distinguish between the scenario where charge billing data is received for an (authorized) session of $EV_1$ and $EV_2$ each and the scenario where charge billing data is received for two (authorized) session of $EV_1$. Charge session may be at the same or different locations to show that linkability across locations (i.e., traceability) is not possible.

*Non-linkability.* Regarding the non-linkability of EV users (PR2), we verify unlinkability of EVs/users based on their authentication/authorization data (as generated by the EV). Analogously to non-traceability, we show that the

(*Honest-but-Curious*) adversary cannot distinguish between a scenario with two authorizations of different EVs and a scenario with two authorizations of the same EV.

### 7.3 Discussion of Requirements

The functional requirements are addressed by the concept design as follows: Credential installation and charge authorization are still possible without user interaction FR1, which ensures user-friendliness. Contract authentication via SSI can be negotiated via the *ServiceDiscoveryReq/-Res* messages FR2. All SSI roles are covered by actors from the ISO 15118 ecosystem as discussed in Sect. 5.1 FR3. Vehicles manage their contract credential in their own wallet FR4. All contract credentials contain the same core elements as discussed in Sect. 5, which allows a CP to authenticate the contract of different eMSPs FR5. Credential installation can be delayed until the first charging session FR6 using the messages described in Sect. 5.3. Communication of the EV (e.g., for credential installation or reading data of the ledger) can still be tunneled via the CP FR7 using the same concepts as for the default ISO 15118 method (e.g., credential installation messages are simply forwarded to the backend in Base64 encoding via OCPP 2.0 [38]). We judge the additional overhead to be acceptable FR8 as discussed in Sect. 7.1.

The security requirements SR1–SR4 are addressed as discussed in Sect. 7.2. In short, the security requirements are shown to be met via symbolic proofs using the Tamarin tool. The corresponding models for automated proof generation are provided online. (See footnote 10) All properties are verified in roughly 30 min on a standard laptop.[11] The published repository includes the defined model/lemmas, the used oracles (for performance such that the model analysis terminates within a reasonable time frame), and instructions on running the models (for reproducibility of the formal analysis).

Analogously, the privacy requirements PR1 and PR2 are addressed as discussed in Sect. 7.2 and the models for automated proof generation are provided online. (See footnote 10) The concept primarily prevents linkability/traceability through the authentication process at the CPO/CP. However, since traditional payment channels are still supported and thus charging sessions must be associated by the eMSP with the respective customers, the eMSP can still link them. This could be fixed via anonymous payment methods, which is out-of-scope for this paper. Additionally, since we focus on the application layer authorization mechanism, linkability based on communication meta data is not addressed by the presented solution. For example, a CPO/CP could potentially track specific EVs based on their MAC addresses, which could be prevented by generic solutions such as MAC address randomization (which is already used by some EV OEMs such as Volkswagen Group [23]). Furthermore, since colluding operators are excluded in the adversary model (cf. Sect. 4.2), the privacy guarantees can be violated if the respective actors collude (e.g., collusion between an eMSP and a CPO to link charge sessions to a location). Colluding operators are outside

[11] Using a Lenovo ThinkPad T14 Gen 1 with 16GB RAM.

the scope of this paper (which focuses on a privacy-by-design solution for charge authorization to minimize privacy risks) but may for example be enforced by regulations.

## 8 Conclusion

In this paper, we propose an approach for using SSIs as trusted credentials for EV charging authentication and authorization in ISO 15118. By using verifiable credentials with zero-knowledge proofs, our solution addresses the privacy problems of ISO 15118 providing unlinkability of charging sessions. Furthermore, our solution uses a decentralized distributed ledger and does not require a complex centralized PKI anymore. Our prototypical implementation and performance evaluation show that the computational and communication overhead of our solution is relatively low and should be acceptable for a real-world implementation. Our formal analysis using Tamarin shows that all required security and privacy properties hold, i.e., still guarantee authentication properties between different actors while preserving the EV user's privacy to the highest possible extent (only eMSP can link a user's charging events for billing purposes). Future work could expand our concept to the authentication of all PnC actors, especially CPs.

**Acknowledgements.** This research work has been partly 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 and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 503329135.

## References

1. Bradbury, M., Taylor, P., Atmaca, U.I., Maple, C., Griffiths, N.: Privacy challenges with protecting live vehicular location context. IEEE Access **8**, 207465–207484 (2020)
2. ChargePoint, DigiCert, Eonti: Practical considerations for implementation and scaling iso 15118 into a secure ev charging ecosystem, May 2019. https://www.chargepoint.com/files/15118whitepaper.pdf
3. Delaune, S., Hirschi, L.: A survey of symbolic methods for establishing equivalence-based properties in cryptographic protocols. J. Log. Algebraic Methods Programm. **87**, 127–144 (2017)
4. DIN Standards Committee Road Vehicle Engineering: Electromobility - Digital communication between a d.c. EV charging station and an electric vehicle for control of d.c. charging in the Combined Charging System. DIN SPEC 70121:2014–12, Deutsches Institut für Normung (DIN) (12 2014)
5. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
6. ElaadNL: EV related protocol study, January 2017. https://www.elaad.nl/research/ev-related-protocol-study/

7. ElaadNL: Exploring the public key infrastructure for iso 15118 in the ev charging ecosystem, November 2018. https://www.elaad.nl/news/publication-exploring-the-public-key-infrastructure-for-iso-15118-in-the-ev-charging-ecosystem/
8. Falk, R., Fries, S.: Electric vehicle charging infrastructure security considerations and approaches. In: Proceedings of INTERNET, pp. 58–64 (2012)
9. Fazouane, M., Kopp, H., van der Heijden, R.W., Le Métayer, D., Kargl, F.: Formal verification of privacy properties in electric vehicle charging. In: Piessens, F., Caballero, J., Bielova, N. (eds.) ESSoS 2015. LNCS, vol. 8978, pp. 17–33. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15618-7_2
10. Firoozjaei, M.D., Ghorbani, A., Kim, H., Song, J.: EVChain: a blockchain-based credit sharing in electric vehicles charging. In: 2019 17th International Conference on Privacy, Security and Trust (PST). IEEE, August 2019. https://doi.org/10.1109/PST47121.2019.8949026
11. Foundation, H.: Getting started with libvcx. https://github.com/hyperledger/indy-sdk/blob/master/vcx/docs/getting-started/getting-started.md (2021). Accessed 28 Feb 2023
12. Foundation, H.: Hyperledger aries rfc 0036 (2021). https://github.com/hyperledger/aries-rfcs/blob/main/features/0036-issue-credential/README.md. Accessed 28 Feb 2023
13. Fuchs, A., Kern, D., Krauß, C., Zhdanova, M.: HIP: HSM-based identities for plug-and-charge. In: Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES 2020, Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3407023.3407066. https://doi.org/10.1145/3407023.3407066
14. Fuchs, A., Kern, D., Krauß, C., Zhdanova, M.: Securing electric vehicle charging systems through component binding. In: Casimiro, A., Ortmeier, F., Bitsch, F., Ferreira, P. (eds.) SAFECOMP 2020. LNCS, vol. 12234, pp. 387–401. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54549-9_26
15. Fuchs, A., Kern, D., Krauß, C., Zhdanova, M.: 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
16. Fuchs, A., Kern, D., Krauß, C., Zhdanova, M., Heddergott, R.: HIP-20: Integration of vehicle-HSM-generated credentials into plug-and-charge infrastructure. In: Computer Science in Cars Symposium. CSCS '20, Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3385958.3430483, https://doi.org/10.1145/3385958.3430483
17. Han, W., Xiao, Y.: Privacy preservation for V2G networks in smart grid: a survey. Comput. Commun. **91**, 17–28 (2016)
18. Hoess, A., Roth, T., Sedlmeir, J., Fridgen, G., Rieger, A.: With or without blockchain? towards a decentralized, ssi-based eroaming architecture. In: Hawaii International Conference on System Sciences (2022)
19. Höfer, C., Petit, J., Schmidt, R., Kargl, F.: Popcorn: privacy-preserving charging for emobility. In: Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles, pp. 37–48 (2013)
20. IEC: Electric vehicle conductive charging system - Part 1: General requirements. IEC Standard 61851–1:2017, International Electrotechnical Commission (2017)
21. ISO/IEC: Road vehicles - Vehicle-to-Grid Communication Interface - Part 2: Network and application protocol requirements. ISO Standard 15118–2:2014, ISO, Geneva, Switzerland, April 2014

22. ISO/IEC: Road vehicles - vehicle-to-grid communication interface - part 2: Network and application protocol requirements. ISO/DIS 15118–2:2018, International Organization for Standardization, Geneva, Switzerland, December 2018
23. Kaiser, C.: Plug in and charge aka autocharge aka plug & charge (2022). https://www.linkedin.com/pulse/plug-charge-aka-autocharge-chris-kaiser. Accessed 26 Sept 20213
24. Kern, D., Krauß, C.: Analysis of e-mobility-based threats to power grid resilience. In: Proceedings of the 5th ACM Computer Science in Cars Symposium, pp. 1–12 (2021)
25. Kern, D., Krauß, C.: Detection of e-mobility-based attacks on the power grid. In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 352–365. IEEE (2023)
26. Kern, D., Krauß, C., Lauser, T., Alnahawi, N., Wiesmaier, A., Niederhagen, R.: Quantumcharge: Post-quantum cryptography for electric vehicle charging. In: International Conference on Applied Cryptography and Network Security, pp. 85–111. Springer (2023).https://doi.org/10.1007/978-3-031-33491-7_4
27. Kern, D., Lauser, T., Krauß, C.: Integrating privacy into the electric vehicle charging architecture. Proc. Privacy Enhancing Technol. **3**, 140–158 (2022)
28. Khovratovich, D., Lodder, M.: Anonymous credentials with type-3 revocation (2018). https://github.com/hyperledger/indy-crypto/blob/master/libindy-crypto/docs/AnonCred.pdf
29. Knirsch, F., Unterweger, A., Engel, D.: Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. Comput. Sci.-Res. Dev. **33**(1–2), 71–79 (2018)
30. Langer, L., Skopik, F., Kienesberger, G., Li, Q.: Privacy issues of smart e-mobility. In: IECON 2013–39th Annual Conference of the IEEE Industrial Electronics Society, pp. 6682–6687. IEEE (2013)
31. Li, H., Dan, G., Nahrstedt, K.: Portunes: privacy-preserving fast authentication for dynamic electric vehicle charging. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 920–925. IEEE (2014)
32. Li, H., Dán, G., Nahrstedt, K.: Portunes+: privacy-preserving fast authentication for dynamic electric vehicle charging. IEEE Trans. Smart Grid **8**(5), 2305–2313 (2016)
33. Lowe, G.: A hierarchy of authentication specifications. In: Proceedings 10th Computer Security Foundations Workshop, pp. 31–43. IEEE (1997)
34. Lux, Z.A., Thatmann, D., Zickau, S., Beierle, F.: Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials. In: 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS). IEEE, Sept 2020. https://doi.org/10.1109/BRAINS49436.2020.9223292
35. Meier, S., Schmidt, B., Cremers, C., Basin, D.: The TAMARIN prover for the symbolic analysis of security protocols. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 696–701. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_48
36. Monteuuis, J.P., Petit, J., Zhang, J., Labiod, H., Mafrica, S., Servel, A.: Attacker model for connected and automated vehicles. In: ACM COMPUTER SCIENCE IN CARS SYMPOSIUM (2018)
37. Mustafa, M.A., Zhang, N., Kalogridis, G., Fan, Z.: Roaming electric vehicle charging and billing: an anonymous multi-user protocol. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 939–945. IEEE (2014)

38. OCA: Open Charge Point Protocol 2.0.1 - Part 2 - Specification. Open standard, Open Charge Alliance, Arnhem, Netherlands, March 2020. https://www.openchargealliance.org/protocols/ocpp-201/
39. Open Fastcharging Alliancey: Autocharge (2017). https://github.com/openfastchargingalliance/openfastchargingalliance/blob/master/autocharge-final.pdf. Accessed 27 Sept 2023
40. Paverd, A., Martin, A., Brown, I.: Modelling and automatically analysing privacy properties for honest-but-curious adversaries. Technical report (2014)
41. Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M.: Decentralized Identifiers (DIDs) v1.0 W3C Candidate Recommendation Draft, May 2021
42. Richter, D., Anke, J.: Exploring potential impacts of self-sovereign identity on smart service systems: an analysis of electric vehicle charging services. In: Business Information Systems, pp. 105–116 (2021)
43. Sabadello, M., et al.: Introduction to DID Auth. Rebooting the Web of Trust VI, July 2018
44. Sporny, M., Longley, D., Chadwick, D.: Verifiable Credentials Data Model v1.1. (2021). https://w3.org/TR/vc-data-model/. Accessed 23 Nov 2021
45. V2G Clarity: RISE-V2G (2017). https://github.com/SwitchEV/RISE-V2G. Accessed 29 Nov 2021
46. 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, December 2019
47. Won, J., Singla, A., Bertino, E., Bollella, G.: Decentralized public key infrastructure for internet-of-things. In: MILCOM 2018–2018 IEEE Military Communications Conference (MILCOM), pp. 907–913. IEEE (2018)
48. Xu, S., Chen, X., He, Y.: EVchain: an anonymous blockchain-based system for charging-connected electric vehicles. Tsinghua Sci. Technol. **26**(6), December 2021. https://doi.org/10.26599/TST.2020.9010043
49. Zelle, D., Springer, M., Zhdanova, M., Krauß, C.: Anonymous charging and billing of electric vehicles. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, pp. 1–10 (2018)
50. Zhao, T., Zhang, C., Wei, L., Zhang, Y.: A secure and privacy-preserving payment system for electric vehicles. In: 2015 IEEE International Conference on Communications (ICC), pp. 7280–7285. IEEE (2015)