Enhancing Anonymity for Electric Vehicles in the ISO 15118 Plug-and-Charge

Nethmi Hettiarachchi[a], Kalikinkar Mandal[b] and Saqib Hakak[c]
Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

Keywords: Electric Vehicle Charging, Plug and Charge, ISO15118, Anonymous Authentication, Authenticated Key Agreement (AKA).

Abstract: ISO 15118 is a standard protocol family that enables the plug-and-charge (PnC) functionality in the electric vehicle (EV) charging architecture. To initiate a charging session, an EV must first authenticate to the charging point (CP) by establishing a TLS connection using its X.509 certificate, followed by authorisation and billing at the end of charging. In this work, we first analyse the privacy of EVs with respect to the information exchanged during the ISO 15118 authentication, charging authorization and billing procedure. We discovered a significant privacy leakage in the current standard, where the initial authentication phase and the billing expose sensitive information that enables various attacks such as charging session linking, EV fingerprinting, profiling and resumption attacks against EV. To address this privacy issue, we first propose an efficient mutual authentication protocol for ISO 15118 PnC that protects the privacy of EVs, including identity and location, against the CP. We analyse the security of our protocol using the Tamarin formal verification tool. The protocol is implemented with various standardised cryptographic schemes and evaluated on different device platforms.

1 INTRODUCTION

Global EV adoption is rapidly increasing due to lower costs, emerging technologies, and government incentives. In 2023, 14 million new EVs were registered, bringing the total to 40 million (International Energy Agency, 2023). To support this growth, new charging strategies and protocols are being developed. ISO 15118 facilitates EV integration into the smart grid, with Edition 1 being widely used and Edition 2 (ISO 15118-20) introducing enhancements. The protocol automates authentication, authorisation, and billing via PnC, eliminating the need for manual payment methods. EVs authenticate using contract credentials from e-Mobility Service Providers (eMSPs), with a valid *eMAID* linked to a billing account. Despite its convenience, ISO 15118 has privacy concerns. It relies on a complex public key infrastructure (PKI) requiring multiple entities and X.509 certificates. EVs must share sensitive information, including provisioning certificates and personally identifiable information (PII), with the CP and CPO for authorization. The charge detail record (CDR) further exposes data, enabling profiling and tracking (Regulation, 2016). CPs can monitor EV charging habits and potentially monetize this data. Any exposed information falls under the EU's GDPR and requires privacy-preserving handling.

Our research focuses on three lacking privacy aspects in ISO 15118: (i) EV identity confidentiality, (ii) location privacy, and (iii) EV untraceability. AKA protocols in 3G, 4G, and 5G are crucial for addressing these concerns. They protect mobile subscribers' identity and location confidentiality, ensuring that authentication occurs without revealing sensitive information to untrusted third parties. AKA enables mobility services even outside regular provider coverage by issuing temporary credentials, preserving user privacy during authentication. Thus we use an AKA protocol-inspired approach to enhancing anonymity for ISO 15118 PnC.

Our Contributions. Our contributions in this paper are two-fold. First, we perform a privacy analysis of the existing ISO 15118 PnC mechanism. Next, to address such privacy leakage, we propose an efficient mutual authentication protocol, providing EV anonymity, along with its security and implementa-

[a] https://orcid.org/0000-0002-0532-1263
[b] https://orcid.org/0000-0002-8228-5016
[c] https://orcid.org/0000-0002-8718-0336

Hettiarachchi, N., Mandal, K. and Hakak, S.
Enhancing Anonymity for Electric Vehicles in the ISO 15118 Plug-and-Charge.
DOI: 10.5220/0013571700003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 475-482
ISBN: 978-989-758-760-3; ISSN: 2184-7711

tions in the ISO 15118 framework. We summarise the key contributions below.

1. We conduct a comprehensive privacy analysis of the ISO 15118 protocol and identify critical privacy leakages and potential attacks. Next, based on the LINDDUN privacy model, we deduce the essential privacy requirements for a secure and privacy-preserving PnC session.
2. To mitigate the privacy leakage, we propose an anonymous mutual authentication protocol that establishes a secure session key for PnC charging.
3. We analyze the security of our protocol using the Tamarin, providing a rigorous security proof to ensure its robustness against potential attacks.
4. We implement our protocol in the EcoG-io/ISO 15118 framework using a Raspberry Pi 4B device and a desktop. We benchmark our protocol for a wide-variety of AKA algorithms Milenage (AES-based) and TUAK (Keccak-based) and cryptographic primitives.

2 RELATED WORK

(Yue et al., 2024) proposed the first anonymous payment scheme for V2G networks, combining blockchain with PBFT consensus, one-time signatures (OTS) (Zaverucha and Stinson, 2010), and bilinear pairing to enable lightweight, anonymous, and binding payments. Direct Anonymous Attestation (DAA) enables privacy-preserving platform authentication (Brickell et al., 2004), and recent PnC authentication protocols (Zelle et al., 2018) and (Zhao et al., 2015) use TPM-based DAA for EV authentication. (Kern et al., 2022) also proposed a TPM-based DAA scheme for EV authentication, but it has several limitations. It requires EVs to be equipped with TPMs, creating hardware dependencies and limiting compatibility with current infrastructure. CPs and CPOs must support DAA verification under high session loads, introducing performance bottlenecks. Additionally, the protocol replaces standard contract certificates with DAA-based X.509 certificates, disrupting the existing PKI model and requiring eMSPs to issue new credentials. These constraints, including a lack of standardisation and TPM requirements at both EV and backend, raise scalability concerns and limit practical deployment within ISO 15118 systems.

3 BACKGROUND

The transition to electrified transportation demands a dedicated charging infrastructure that is efficient, user-friendly, and interoperable. The e-mobility architecture enables interoperability by allowing EVs to charge with any provider through physical compatibility and standardized protocols like ISO 15118 for secure authentication, authorization, and billing. Key entities include EVs, CPs, CPOs, eMSPs, Certificate Provisioning Services (CPS), and grid operators, which communicate via standardized protocols to manage charging sessions (see Figure 1). An EV contains an EVCC for communication, while the SECC acts as the CP interface. EV-CP communication occurs over a secure TLS channel using ISO 15118, with Edition 1 using TLS 1.2 and Edition 2 mandating TLS 1.3 for mutual authentication via manufacturer-issued certificates. It recommends `TLS_AES_256_GCM_SHA384` or `TLS_CHACHA20_POLY1305_SHA256` cipher suites. Backend communication with eMSP uses OCPI and OCPP over TLS.

Provisioning Credential Certificate. During EV production, the OEM generates provisioning credentials, including a PCID (which uniquely identifys the EV for billing and other purposes), a provisioning certificate (PC_{cert}), and a key pair (PC_{sk}, PC_{pk}). PC_{cert} contains the PCID and the EV's public key and is digitally signed by the OEM's Certificate Authority (OEM-CA) to ensure authenticity and integrity. This certificate enables the EV to authenticate with charging stations and service providers during its initial charging session and obtain the CC_{cert}, the long-term certificate for charging authorization.

Obtaining the Contract Certificate and Installation. To acquire PnC services, the EV must have a valid charging contract with an eMSP. The EV owner presents the PCID to the eMSP, which links it to a billing account identified by an *eMAID* (Electric Mobility Account Identifier) as specified in ISO15118. The eMSP generates a key pair (CC_{pk}, CC_{sk}) and a contract certificate (CC_{cert}) containing the *eMAID*. Figure 1 shows the process of obtaining and installing the contract certificate. During the first charging session, (CC_{cert}) and keys (CC_{pk}, CC_{sk}) are installed into the EV over a secure TLS channel. The EV sends a credential installation request (containing PC_{cert} signed with PC_{sk}) to the CP, which forwards it to the eMSP via CPS. The eMSP encrypts $CC_{sk}$with PC_{pk} from PC_{cert}, and CPS signs the response after verifying the contract certificate chain. The EV, trusting CPS, decrypts the response using PC_{sk} to retrieve $CC_{sk}$and stores the credentials securely for future PnC authentications.

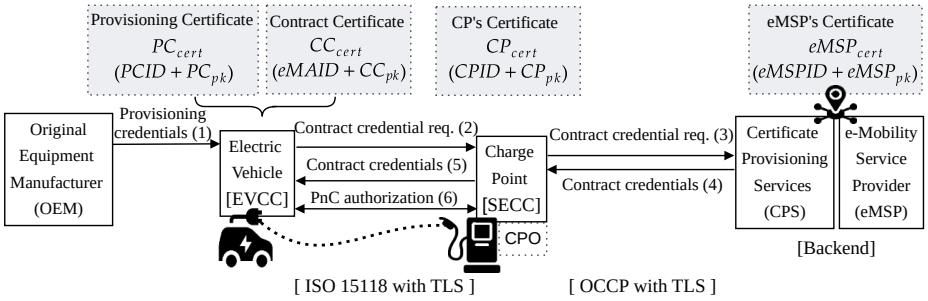

Figure 1: eMobility architecture.

Table 1: Summary of EV Privacy Attacks in ISO 15118 Charging Networks.

Attack	Description and Impact
A1: Charging Session Linking Attack	TLS 1.3 session resumption improves efficiency but exposes EV information through resumption tickets. A passive adversary can perform linking EV sessions by correlating reused tokens, leading to metadata exposure such as charging times and session frequency. CPs or adversaries can also build behavioural profiles by profiling interaction history across networks (Arfaoui et al., 2019).
A2: EV Fingerprinting and Profiling Attack	During the TLS handshake, EVs expose protocol parameters that allow adversaries to generate unique fingerprints (e.g., JARM, JA3 (Althouse et al., 2017a), JA3S (Althouse et al., 2017b)). This enables EV tracking and profiling across a chain of CPs (belongs to one company), even with session resumption, and metadata exposure revealing the EV model, OS, and firmware version.
A3: Charging Point Surveillance and EV Profiling	Although TLS 1.3 encrypts certificates, CPs can still access EV certificates upon receiving. This permits linking EV sessions across locations, metadata exposure of cryptographic configurations, and monetizing EV data through sales to manufacturers or third parties. Additionally, tracking movement can reveal sensitive locations such as home or workplace, and malicious CP attacks can impersonate EVs or exploit their credentials.

Using the Contract Credentials. When an EV plugs into the CP for charging, ISO15118 manages the process without customer involvement. Before communication, TLS 1.3 mutual authentication is performed: the CP uses its SECC certificate, and the EV uses its manufacturer-issued vehicle certificate. Over the TLS channel, the EV sends its contract credentials (CC_{sk}, *eMAID*) to the CP for authorization. The CP validates the credentials, authorizes the EV, and initiates charging. During the session, the EV and CP periodically exchange signed charging status and meter reading messages, with the EV signing using CC_{sk}. A detailed description of ISO15118 cryptographic operations follows in the next section.

ISO15118 PnC Charging Authorisation. The ISO15118 communication protocol builds upon IEC 61851, which defines basic signalling and voltage levels for EV charging. They operate together during a session. When the EV plugs the charging cable into the CP, the process initially follows IEC61851. The control pilot line, a dedicated wire in the cable, connects the EV and CP and transmits PWM signals. A PWM duty cycle of 5% signals both parties to initiate ISO15118 communication. ISO15118 uses certificate-based mutual authentication. To enable plug-and-charge, the EV must have an installed contract certificate. The EV submits its CC_{cert} and the eMSP certificate chain to the CP, which verifies the eMSP chain against a locally installed root and validates the CC_{cert}. The CP then sends its CPID and a challenge (fresh nonce) to the EV. To prove authenticity, the EV signs and returns the eMAID (CPID and nonce) and the challenge. The CP verifies the signature using $CC_{pk}$from CC_{cert}. Upon successful verification, the EV is authorized and charging begins.

4 PRIVACY ASSESSMENT OF THE ISO 15118 PnC PROTOCOL

Privacy Attacks on ISO 15118 PnC

Security and Privacy Requirements. To define privacy and security requirements, we adopt the LINDDUN framework (Deng et al., 2011), known for its structured methodology and privacy threat trees aligned with LINDDUN types (Sion et al., 2018). This approach identifies risks and derives requirements in environments with *honest but curious* entities; we model CCP and CPO as such. We also consider an external attacker or passive eavesdropper monitoring PnC traffic to infer EV locations, move-

Table 2: LINDDUN Privacy Requirements for ISO 15118.

Privacy Requirement	Acron.	LINDDUN Category
EV identity confidentiality	P1	Identifiability, Linkability
Prevention of unique identification	P2	Identifiability, Linkability, Detectability
Charging session unlinkability	P3	Linkability, Disclosure of Information
Charging data aggregation prevention	P4	Linkability, Disclosure of Information, Non-repudiation
EV location privacy	P5	Linkability, Disclosure of Information

ment patterns, and session correlations, posing risks like tracking EV owners or revealing home and work locations. Based on these adversaries and LINDDUN threat modeling, privacy requirements are summarized in Table 2.

5 OUR THREE-PARTY AUTHENTICATION PROTOCOL

Here, we present our protocol step by step with the aid of detailed protocol diagrams. Each diagram illustrates both the cryptographic operations and the corresponding message flow within the ISO 15118 protocol. To enhance clarity, the ISO 15118 message names are provided beneath the arrows in the diagrams, allowing readers to easily map each step to the standard protocol sequence.

Cryptographic Credentials. Like ISO 15118, we assume that each EV holds two pairs of certificates: 1) PC_{cert} for the digital signature key pair (PC_{pk}, PC_{sk}) and an EV ID: *PCID*, received from OME; and another certificate CC_{cert} for (CC_{sk},CC_{pk}) and *eMAID* received from the eMSP for PnC. The CP also holds CP_{cert} for (CP_{sk},CP_{pk}) and *CPID*. The eMSP has a root certificate $eMSP_{root}$ and its own certificate $eMSP_{cert}$ for the signature key $(eMSP_{pk}, eMSP_{sk})$ and *eMSPID*. We also require the eMSP to have a certificate for long-term/static public-private key pair (PK_{eMSP}, SK_{eMSP}) for the public-key encryption (PKE) used in the key-encapsulation mechanism.

EV Authenticating CP Messages $(EV \Rightarrow CP)$**:** It is a standard certificate-based challenge-response protocol. We present it in Figure 2 for completeness. The EV randomly samples a number n_{EV} and sends it to the CP along with a certificate request. After receiving n_{EV}, the CP signs n_{EV} and its ID *CPID* with the key CP_{sk} and sends σ_{CP}, *CPID*, and CP_{cert} to the EV. Upon receiving, the EV verifies σ_{CP} by extracting CP_{pk} from the received CP_{cert}. If the verification is successful, the EV proceed to the next step, otherwise, it aborts.

EV and eMSP Mutual Authentication Messages (EV $\Leftrightarrow$ eMSP)**:** The communication between the EV and eMSP is done via the CP. The key idea of our protocol is that the EV sends a random challenge and its certificate to the eMSP in the encrypted form so that the CP does not have any information about the EV's identity and the EV and eMSP establish a shared session key that is used for the authentication and key establishment between the EV and CP. Figure 3 illustrates the three-party protocol for generating and verifying authentication vectors. The eMSP authenticates and establishes a shared key with the EV as follows:

(a) eMSP Authenticates EV and Computes a Shared Key (eMSP $\Rightarrow$ EV)**:** The EV runs a key encapsulation using the eMSP's public key PK_{eMSP} to obtain a key K_s and ciphertext c. It saves K_s for authentication, signs c using CC_{sk} to produce σ_{EV}, and encrypts CC_{cert} and a challenge n_{EV} with authenticated encryption to get ciphertext and tag C. The EV sends (c, σ_{EV}, C) to the CP, who forwards it to the eMSP via a secure channel (e.g., TLS). Upon receiving (c, σ_{EV}, C), the eMSP decapsulates c using SK_{eMSP} to recover K_s, decrypts C to retrieve $D = CC_{cert} | n_{EV}$, and extracts CC_{pk} from CC_{cert} to verify σ_{EV}. If verification succeeds, K_s is accepted. The EV and eMSP then derive a shared secret K from K_s, n_{EV}, n_{eMSP}, *eMAID*, and *eMSPID* using a key derivation function.

(b) eMSP Computes an Authentication Vector and Keys: To enable CP authentication without revealing the EV's identity or certificate, the eMSP generates an authentication vector from K_s and a random nonce R, following the 3GPP AKA protocol (3rd Generation Partnership Project (3GPP), 2021). Both EV and eMSP maintain counters, SQN_{EV} and SQN_{eMSP}, to prevent replay attacks. Using one-way keyed functions f_1 to f_5 (3rd Generation Partnership Project (3GPP), 2024), the eMSP computes authentication parameters: $MAC = f_1(SQN_{eMSP}, R, K)$ and $xRES = f_2(R, K)$. Sequence number protection is ensured via an anonymity key $AK = f_5(R, K)$, and the authentication vector is $AUTH = (MAC, CONC)$ with $CONC = SQN_{eMSP} \oplus AK$ and $xRES$. Encryption key CK and integrity key IK are derived using

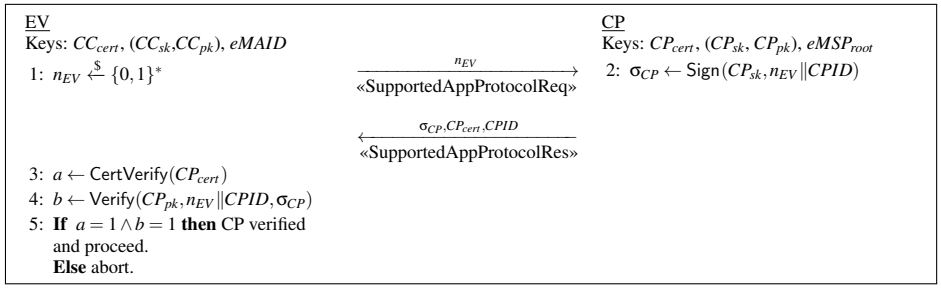

Figure 2: EV authenticates the CP using a certificate.

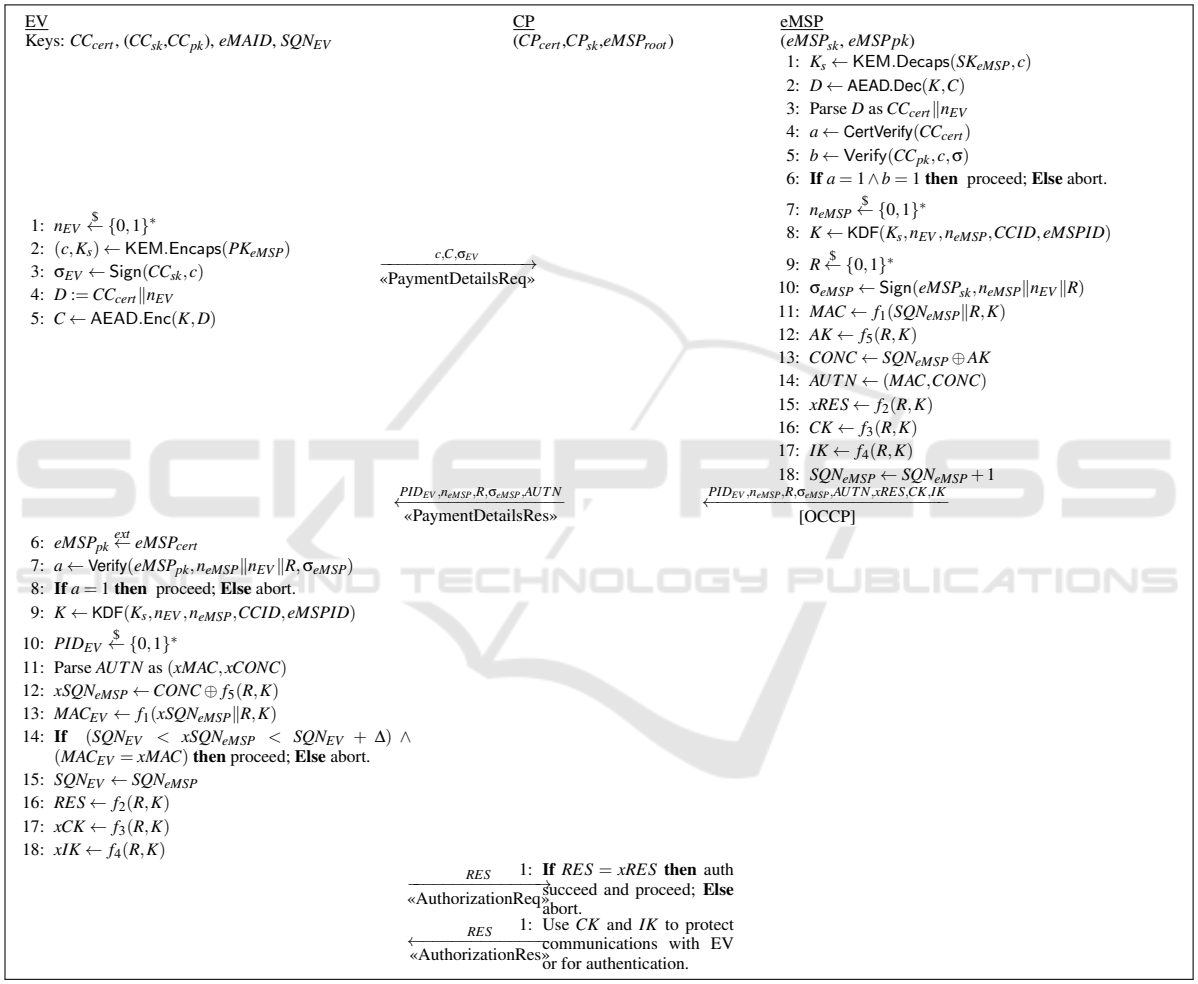

Figure 3: Three-party protocol for generating and verifying authentication vectors.

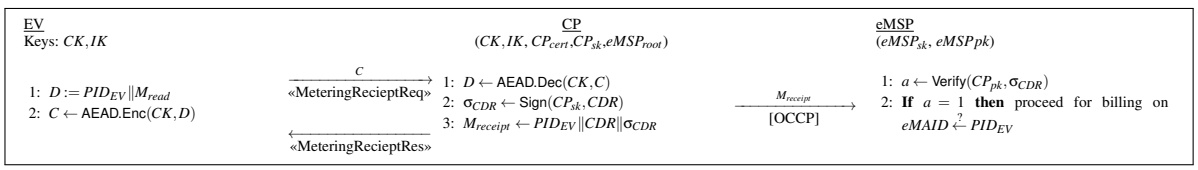

Figure 4: Billing process involving EV, CP, and eMSP.

$f_3(R,K)$ and $f_4(R,K)$, respectively. The eMSP signs $n_{eMSP}|n_{EV}|R$ using $eMSP_{sk}$ to produce σ_{eMSP}, then sends $n_{eMSP},R,\sigma_{eMSP},(AUTH,xRES)$ and (CK,IK) to the CP via a secure channel. The CP stores $xRES$, CK, and IK, and forwards $n_{eMSP},R,\sigma_{eMSP},AUTH$ to the EV.

(c) EV Authenticates eMSP and Computes an Authentication Vector and Keys (EV $\Rightarrow$ eMSP)**:** Upon receiving $(n_{eMSP},R,\sigma_{eMSP},AUTH)$ from the CP, the EV verifies n_{eMSP} and R using σ_{eMSP} and $eMSP_{pk}$. If successful, it computes the shared K using the KDF. The EV extracts $xCONC$ and $xMAC$ from $AUTH$ and checks: (i) if $xMAC$ matches the locally computed MAC from K, R, and recovered SQN_{eMSP}; and (ii) if $xSQN_{eMSP}$; satisfies $xSQN_{eMSP} > SQN_{EV}$; and $xSQN_{eMSP} < SQN_{EV}+\Delta$, where Δ is a threshold. If both hold, the EV updates SQN_{EV}; and derives RES, xCK, and xIK using K and R.

(d) CP Authenticates EV for Charging (CP $\Rightarrow$ EV)**:** When the CP sends an authentication request, denoted by `Auth_Req`, the EV computes RES using f_2 on R and K and sends it to the CP. After that, the CP checks whether the received RES equals $xRES$ received from the eMSP. If they are equal, the CP successfully authenticates the EV and authorizes the EV for charging. The rest of the communications such as the current meter reading between the EV and CP are (confidentiality and integrity) protected using the keys xCK and xIK.

EV Sends Metering Data to eMSP for Billing (EV $\longleftrightarrow$ CP)**:** The EV prepares the metering data by concatenating its pseudonymous ID (PID_{EV}) with the meter reading (M_{read}). To ensure confidentiality and integrity, the EV encrypts this data using an AEAD encryption scheme, producing the ciphertext C. The encrypted metering data is then transmitted to the CP. Upon receiving C, the CP decrypts it to retrieve the original metering information. The CP then generates a Charging Data Record (CDR) and signs it using its private key (CP_{sk}) to ensure authenticity. The signed CDR, along with the EV's pseudonymous ID, is compiled into a metering receipt ($M_{receipt}$), which is then forwarded to the eMSP. The eMSP verifies the signature on the CDR using the CP's public key (CP_{pk}). If the verification succeeds, the eMSP links the session to the EV's contract ID ($eMAID$) and proceeds with the billing process on the corresponding user. Figure 4 illustrates the secure transmission of meter reading data to the eMSP for billing, ensuring data confidentiality, integrity, and authenticity throughout the process.

6 SECURITY ANALYSIS

Our study targets three objectives: EV confidentiality, location privacy, and EV unlinkability by the CP. Using the LINDDUN framework, we derived privacy requirements (P1–P5) and verified them with the Tamarin Prover (Meier et al., 2013). Lemma 1 proves injective agreement between EV and CP to ensure authentication uniqueness, while Lemma 2 confirms the encrypted CCID (eMAID) is never exposed to the CP.

Data: All traces
For all EV, CP, ds, and #i:;
  If `Commit`(EV, CP, ds) happens at step #i:;
    There exists a step #j such that:;
      `Running`(CP, EV, ds) happens at step #j,;
      #$j < i$, and;
      For all steps #i_2, if
        `Commit`(EV, CP, ds) happens at #i_2,
        then #$i_2 = i$.;

Lemma 1: Injective Agreement Between EV and CP.

Data: All traces
For all EV, CP, C, and #i:;
  If `AEADEncryptionPerformed`(EV, C)
    happens at step #i and
    `CipherSentToCP`(CP, C) happens at step #i:;
      There exists a step #j such that:;
        `AEADDecryptionPerformed`(eMSP, C)
          happens at step #j.;

Lemma 2: CCID (eMAID) Not Shared with CP.

Privacy Properties:

- EV Identity Confidentiality (P1): Prevents CP and CPO from learning the EV's identity ($eMAID$) during authentication; *E*ncrypted CCID Shared with CP Only proves $eMAID$ is AEAD-encrypted and only the eMSP can decrypt it.
- Prevention of Unique Identification (P2): *I*njective Agreement Between EV and CP and *E*ncrypted CCID Shared with CP Only ensure session freshness and that CP cannot learn the $eMAID$ or key material, preventing tracking.
- Charging Session Unlinkability (P3): *E*ncrypted CCID Shared with CP Only guarantees session independence by encrypting $eMAID$, preventing CP from linking multiple sessions.
- Charging Data Aggregation Prevention (P4): *E*ncrypted CCID Shared with CP Only ensures the CPO cannot learn $eMAID$ or aggregate data across locations.
- EV Location Privacy (P5): *E*ncrypted CCID Shared with CP Only prevents CPs from learning

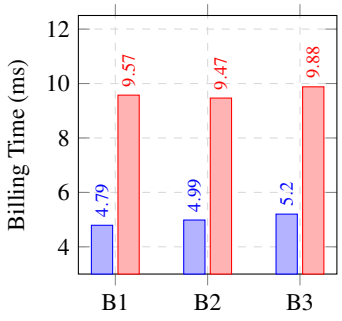

Figure 5: Total Billing Time Comparison on Desktop (blue) and Raspberry Pi (red). B1: AES-GCM, B2: Ascon, B3: ChaCha20-Poly1305.

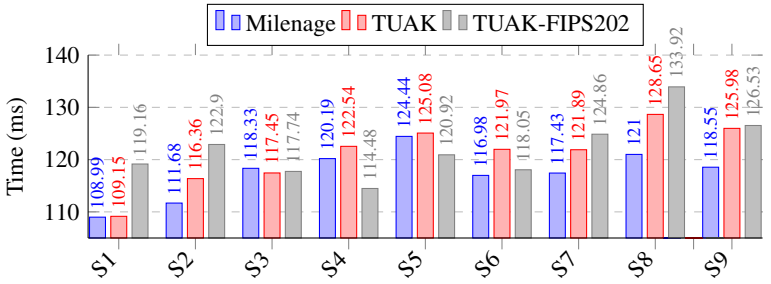

Figure 6: Comparing AKA protocol execution time on Desktop across S1–S9.

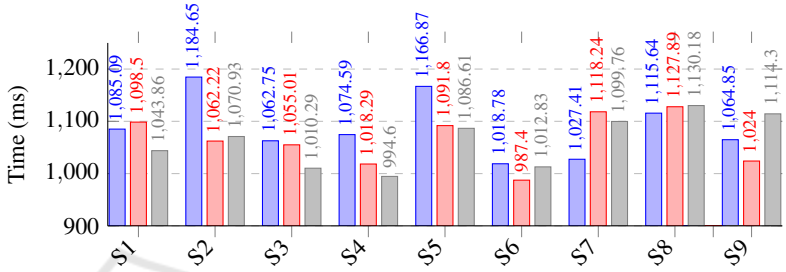

Figure 7: Comparing protocol execution time on Raspberry Pi across S1–S9.

eMAID, ensuring EV movements cannot be correlated by the CPO.

7 EXPERIMENTAL EVALUATION

Testbed Devices and Protocols. We implemented our hybrid authentication protocol in the setup shown in Figure 2 and Figure 3. The EV runs on a Raspberry Pi 4 Model B (2 GB RAM, ARM Cortex-A72, 1.5 GHz), and the CP is deployed on a 64-bit desktop with an Intel Core i7-9700 (3.0 GHz, 16 GB RAM, 8 cores with hyper-threading). To simulate plug-and-charge, we connected the EV and CP via a direct network cable. For simplicity, the eMSP runs in the same environment as the CP, and uses the TLS-based OCPP protocol, which is outside this work's scope.

Security Implementation Details. The hybrid authentication protocol was implemented in C as a standalone cryptographic module, easily integrable into the EcoG-io/ISO 15118 framework (EcoG.io, 2024) using C-type bindings for Python. The protocol is optimized with mbedTLS (Limited, 2024) and OpenSSL 3.0.5 (Project, 2024). It uses the NIST P-256 curve for ECDSA digital signatures, and supports authenticated encryption with ASCON AEAD, AES-GCM, and ChaCha20-Poly1305. HKDF is employed for key derivation, with SHA-256, SHA-3, and ASCON hash as hashing options. For the AKA protocol, we implemented the Milenage set, TUAK set (3rd Generation Partnership Project (3GPP), 2024), and an enhanced TUAK variant using FIPS 202 (Bertoni et al., 2024). We experimented with various combinations to identify efficient cipher suites, particularly targeting lightweight designs for constrained environments. For further discussions, acronyms are used to represent cipher suites. Our cipher suites are:

S1: `AES_GCM_SHA2` S2: `AES_GCM_SHA3`
S3: `AES_GCM_AsconAEAD`
S4: `AsconAEAD_SHA2` S5: `AsconAEAD_SHA3`
S6: `AsconAEAD_AsconHash`
S7: `ChaCha20-Poly1305_SHA2`
S8: `ChaCha20-Poly1305_SHA3`
S9: `ChaCha20-Poly1305_AsconHash`

7.1 Performance Results

This section presents our performance evaluation across two aspects: 1) charging authentication and 2) billing. We conducted micro-benchmarking to measure function-level execution times and macro-benchmarking for overall protocol performance, enabling comparison against the ISO 15118-20 standard. Figure 6 and Figure 7 compares execution times across cipher suites S-S9 on desktop and Raspberry Pi, respectively. Figure 5 depicts the execution time comparison of the billing procedure. Our results show that the Milenage-based AKA protocol with `AES_GCM_SHA2` achieves the best performance on desktop, while the TUAK-based AKA protocol with `Ascon` is optimal for Raspberry Pi. `AES-GCM` is the best choice for billing on desktop, and `Ascon AEAD`

performs best for Raspberry Pi billing. All execution times are within acceptable limits and outperform ISO15118 constraints (International Organization for Standardization, 2022).

8 CONCLUSIONS

The current ISO-15118 PnC architecture lacks inherent privacy protections. As a result, during authentication, charging authorization, and billing, the system exposes a significant amount of personal information about the EV to the CP and CPO. To address this, we propose an anonymous authenticated key establishment protocol for ISO-15118 PnC charging, leveraging the KEM/DEM mechanism, inspired by 3GPP AKA protocols. Our protocol ensures EV identity confidentiality against CP and CPO, location privacy and EV untraceability. It maintains fundamental security properties, is optimised through cross-platform evaluations of computational and energy efficiency, and is formally verified with the Tamarin Prover for robustness. Overall, it provides a secure, scalable, and privacy-preserving enhancement for ISO-15118 PnC.

ACKNOWLEDGEMENTS

The first and second authors are partially funded by the NB Power cybersecurity research chair grant.

REFERENCES

3rd Generation Partnership Project (3GPP) (2021). 3gpp ts 33.535: Authentication and key management for applications (akma). Accessed: 2025-01-22.

3rd Generation Partnership Project (3GPP) (2024). 3g security; security architecture. Technical Specification TS 33.102, 3rd Generation Partnership Project (3GPP). Release 18, Accessed: 2024-11-30.

3rd Generation Partnership Project (3GPP) (2024). *TUAK: A New Set of 3GPP Authentication and Key Generation Algorithms*.

Althouse, J., Atkinson, J., and Atkins, J. (2017a). JA3: Fingerprinting TLS Clients. https://github.com/salesforce/ja3. Accessed: 2024-12-14.

Althouse, J., Atkinson, J., and Atkins, J. (2017b). JA3S: Server-Side TLS Fingerprinting. https://github.com/salesforce/ja3. Accessed: 2024-12-14.

Arfaoui, G., Bultel, X., Fouque, P.-A., Nedelcu, A., and Onete, C. (2019). The privacy of the tls 1.3 protocol. *Cryptology ePrint Archive*.

Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2024). The keccak reference - fips 202: Sha-3 standard: Permutation-based hash and extendable-output functions. https://keccak.team/specifications.html\#FIPS_202.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145.

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

EcoG.io (2024). Iso 15118 framework. https://github.com/EcoG-io/ISO15118.

International Energy Agency (2023). Global ev outlook 2023: Catching up with climate ambitions. Technical report, International Energy Agency (IEA). Accessed: 2024-11-30.

International Organization for Standardization (2022). ISO 15118-20:2022 - Road vehicles – Vehicle to grid communication interface – Part 20: 2nd generation network layer and application layer requirements.

Kern, D., Lauser, T., and Krauß, C. (2022). Integrating privacy into the electric vehicle charging architecture. *Proceedings on Privacy Enhancing Technologies*.

Limited, A. (2024). *mbedTLS: Open Source SSL/TLS Library*. Version 3.2.1.

Meier, S., Schmidt, B., Cremers, C., and Basin, D. (2013). The tamarin prover for the symbolic analysis of security protocols. In *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, pages 696–701. Springer.

Project, T. O. (2024). Openssl: The open source toolkit for ssl/tls. https://www.openssl.org/. Version 3.0.5.

Regulation, P. (2016). Regulation (eu) 2016/679 of the european parliament and of the council. *Regulation (eu)*, 679:2016.

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

Yue, X., Bi, X., Yang, H., Bai, S., and He, Y. (2024). Pap: A privacy-preserving authentication scheme with anonymous payment for v2g networks. *IEEE Transactions on Smart Grid*.

Zaverucha, G. M. and Stinson, D. R. (2010). Short one-time signatures. *Cryptology ePrint Archive*.

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

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