OFFLINE CERTIFICATE VERIFICATION & TRUST IN THE EV-CHARGING PKI

Pol Van Aubel
Radboud University – Netherlands
pol.vanaubel@cs.ru.nl

ABSTRACT

The ISO 15118 protocol – used for communication between Electric Vehicles (EVs) and their charge points – requires a Public Key Infrastructure (PKI). However, actually implementing such a PKI involves additional policy and design choices beyond what ISO 15118 specifies. The German VDE Association for Electrical, Electronic & Information Technologies has published application guidelines for certificate handling in the ISO 15118 PKI. Similarly, the Dutch organization ElaadNL has been running a project to design and implement a single PKI for use by the entire EV-charging ecosystem, and published a set of implementation guidelines that clarify the choices they made.

There are two important remaining issues with the ISO 15118 PKI design that are not solved by these implementation guidelines. First, it is not possible to do adequate offline verification of certificates. Second, the separate role of a Certificate Provisioning Service undermines the (security) policies of the PKI. We propose fixes for both issues: the first by additional technical requirements on the information certificates carry, thus enabling a form of offline verification; the second by requiring neutrality on the top level of the PKI.

1. INTRODUCTION

Charging an Electric Vehicle (EV) at a public charge point often requires the driver to present a smart card to the charge point. The purpose of this is to link the driver to an account that can be billed for the energy consumed. But this process is not yet standardized across Europe, with drivers who cross country borders encountering charge points that they cannot use [8, items 38–41].

The ISO 15118 protocol [3] standardizes a mechanism for automating that process, where an EV presents its charging contract to the charge point using cryptographic certificates. This requires a Public Key Infrastructure (PKI). Building a PKI involves making technical and policy decisions about its structure. ISO 15118 leaves some of those decisions open, and several organizations are working on creating a PKI that satisfies ISO 15118. For example, the German VDE Association for Electrical, Electronic & Information Technologies has published application guidelines for certificate handling in the ISO 15118 PKI [9]. These guidelines build upon the requirements of ISO 15118, making explicit decisions about ambiguities left in the standard. Similarly, the Dutch organization ElaadNL is working on a PKI design, and has published their design rationale and guide for implementation of the PKI [4].

There are two important issues with the ISO 15118 PKI that are not adequately addressed by the VDE guidelines nor the ElaadNL guidelines:

1. Offline verification of contract certificates is not reliable.
2. The separate role of a Certificate Provisioning Service introduces security policy enforcement issues.

In Section 2 we will explain in more detail what a PKI is and what PKI design ElaadNL arrives at. In Section 3 we will present a solution for issue 1, and in Section 4 we will argue that issue 2 is best solved by enforcing neutrality at the highest level of the PKI.

2. PKI FOR EV-CHARGING

A PKI is a way to manage cryptographic keys through the use of certificates. The PKI is most often structured as a tree, as depicted in Figure 1, where trust in the validity of leaf certificates is based on a path of signatures to so-called root certificates. The leaf certificates are the ones actually encoding contracts, devices identity, etc.

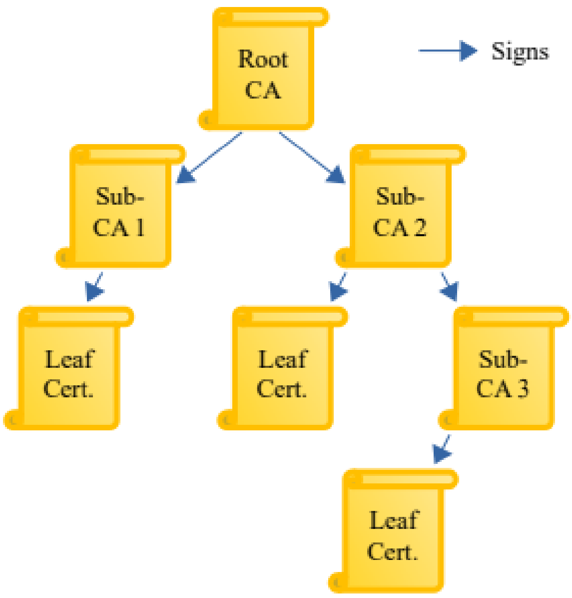

Figure 1: PKI structured as a tree, with signatures from the root CA to the leaf certificates. The trust path follows the signatures back up the tree.

The simplest form of this tree is to have a single Certificate Authority (CA) root certificate, for which the public key is installed in all devices that should trust the root. Any certificate used in the PKI is, through a chain of signatures by intermediate certificates from sub-CAs, linked to the trusted root CA, and thereby deemed trusted as well – the root CA has vouched for it.

As another example, consider the PKI in use for the public Internet, or "Internet PKI". Most people come into contact with this PKI every day, because it is used for securing public websites with TLS. This PKI has many different companies functioning as root CAs. Not all of these are trusted by all software by default. To standardize the decision process about which root CAs are trustworthy, the CAs, browser vendors, operating system vendors, and other interested parties have come together in a voluntary organization, the CA/Browser forum, that publishes and enforces industry guidelines for management of the PKI[1]. Browser vendors have a very strong position in this forum: they do not run root CAs themselves, but they can decide which root CAs to include as trusted in their browsers. This effectively forces all CAs that want to participate in the Internet PKI to comply with the forum's requirements.

Setting up a PKI for the EV-charging ecosystem also means the parties in the ecosystem must agree which root CA to trust. ISO 15118 suggests five root CAs for the entire world, to provide some administrative flexibility (one for each major landmass) while keeping the number of ultimately trusted parties low.

The need to agree on how to decide which root CAs will be trusted and should be installed in EVs is one reason why adoption of a single PKI for the EV-charging ecosystem has, so far, not happened. The PKI design from ElaadNL explains that a root CA must be a neutral party. This is to ensure that there are no conflicts of interest in providing any other actor with a sub-CA certificate, ensuring there is no barrier to entry other than the security requirements applied to all actors. We agree with this requirement, and therefore will assume in the rest of this chapter that a *neutral* party takes the role of the root CA, allowing any actor that conforms to the PKI rules (as laid down by the protocols and the actors in the ecosystem) to join the PKI.

The ElaadNL PKI design has two distinct ways of structuring the relationship from the leaf certificates to the root CA. These ways coexist in this PKI:

1. A Peer-to-Peer structure, where e-Mobility Service Providers (eMSPs), Charge Point Operators (CPOs), etc. are directly underneath the root CA. The path in this case is from contract certificate, to eMSP / CPO sub-CA, to root CA.
2. A centralized structure, where a roaming hub or clearing house functions as a first layer of sub-CA under the root CA, and the eMSPs, CPOs, etc. are provided sub-CA certificates by the clearing house. There is a path from contract certificate, to eMSP / CPO sub-CA, to clearing house sub-CA, to root CA.

1 https://cabforum.org/

The rest of this paper assumes a PKI according to ElaadNL's design is used. As already mentioned, we believe there are two important issues with how this PKI is currently designed:

1. Offline verification of contract certificates is not reliable, which we solve in Section 3.
2. The separate role of a Certificate Provisioning Service introduces security policy enforcement issues and highlights the need for neutral root CAs, explained in Section 4.

3. RELIABLE OFFLINE VERIFICATION OF CONTRACT CERTIFICATES

In this Section we will first elaborate on why offline verification of contract certificates is not reliable in the PKI design for ISO 15118. Then, we will propose a solution by adding a field that an offline check can inspect to all certificates.

Third-party issuance problem for certificates

Client certificates – used to authenticate clients to servers – are a less commonly used type of leaf certificate in PKIs. Most of the PKIs in existence are primarily intended for server-to-client authentication. Integrating client certificates into a public PKI – that is, a PKI for multiple organizations to use freely – brings some additional challenges. Most importantly, whether a client certificate is valid should not merely depend on there being a path to any public *root* CA in the PKI.

Suppose we have two organizations, eMSP **A** and CPO **B**, both with sub-CAs for signing client certificates. These sub-CAs are signed by the same root CA. Suppose servers are configured to accept client certificates that verify up to the root CA. If a client presents a legitimate certificate to a server, specifying it is a client of eMSP **A**, signed by sub-CA **A**, this validates up to the root CA. But if a client presents a certificate to a server, specifying it is a client of eMSP **A**, but signed by sub-CA ***B***, it *also* has a path to the root CA, and hence would be valid if the only requirement is that such a path exists. But **B** should not be issuing client certificates for **A**, and this certificate should be deemed invalid! This is the *third party issuance* problem for client certificates.

The same third party issuance problem exists for ISO 15118 contract certificates: a car with a contract issued by eMSP **A** should have that contract certificate signed by sub-CA **A**, not sub-CA **B**.

Of course, for sub-CA **B** to sign such a client certificate for **A** means that sub-CA **B** is acting maliciously. But this is not a far-fetched scenario. In the public Internet PKI there have been several high-profile cases where Certificate Authorities were compromised and used to issue fraudulent server certificates. Both DigiNotar [5] and Comodo [6] were attacked and compromised in 2011. DigiNotar's compromised CA was used to create, among others, a certificate for `*.google.com` which was accepted by most systems in use at the time, and used in Iran to conduct a man-in-the-middle attack against users connecting to Google services. Interestingly, the Google Chrome web browser did *not* accept these certificates, because Google had started shipping it with additional restrictions on certificates for Google's own websites. Therefore it did not accept those certificates signed by DigiNotar [5]. Both these incidents showed that the classic verification model, where any server certificate is valid as long as it has a valid path to a root CA, has broken down [7]. This too is a third-party issuance problem: certificates were issued by root CAs that were not supposed to for those domains.

Aside from policy changes to ensure better security practices at CAs, technical measures were developed to fix the third-party issuance problem. One such measure is DNS-based Authentication of Named Entities (DANE) which allows a server to communicate through the Domain Name System which CAs are allowed to sign its server certificates. Another is Certificate Transparency (CT), which keeps a public log of all issued server certificates. Browsers can check whether the certificate they are presented with is in the log (and reject any that aren't), and domain administrators can check whether any unexpected parties are issuing certificates for their domains.

In client authentication setups the third party issuance problem is usually solved by not anchoring the trust at a public root CA at all. Instead, a private root CA or one specific sub-CA under the control of the organization using the certificates is used. For example, server systems would be told to trust only sub-CA **A**. But this solution only works if the validity of client certificates only has to be verified by the same organization that issued them. We run into a problem when different organizations have to verify each others' certificates, as is the case for the contract- and client certificates in the EV-charging ecosystem. Contracts from eMSP **A** may be presented to systems run by CPO **B**, and clients from CPO **B** may connect to eMSP **A**. Both eMSP **A** and CPO **B** could simply trust each other's sub-CA for all certificates issued by them, but then we reintroduce the third party issuance problem: eMSP **A** might sign a certificate for CPO **B**, and vice versa. We need a way to verify that a certificate claiming to belong to an organization was indeed issued by a sub-CA from that organization. The ElaadNL guide covers how to do an *online* certificate validity check using the existing mechanisms OCSP and Certificate Revocation Lists [4], but none of these work *offline* to detect third party issuance in a timely fashion. OCSP is an online check, and a Certificate Revocation List, whether online or offline, only contains revoked certificates, so only works to prevent further abuse, not to detect it in the first place. Offline checking is only considered as a backup option, but it is still an important backup to have, because a charge point *must* still function when it temporarily has no network connection. ElaadNL assumes that an offline check would take between 1 month and 2 years to detect fraudulent certificates, which also applies to third party issuance [4].

Alternatives not yet considered in [4] are DANE and CT, but these don't work as an offline detection mechanism either: DANE is online, and CT would require a public log of all issued contract- and client certificates, which is unacceptable from a privacy perspective.

Solving third party issuance with Provider ID

We propose extending an existing requirement from ISO 15118, so that an offline check that instantly detects third party issuance is possible. ISO 15118 contract certificates are currently already required to use the e-Mobility Account Identifier (EMAID) as their Common Name field [3]. The EMAID has a Provider ID, which is a 3-digit alphanumeric code. ISO 15118 suggests this code should be assigned by a central issuing authority such as the eMI[3] group [3,9]. Since that means every actor in this PKI has a unique Provider ID, we simply need to add the requirement that all the intermediate certificates up to the root CA, including the sub-CA certificates, must carry the same Provider ID somewhere in the certificate. The root CA and sub-CAs must ensure that the Provider ID in a certificate they are signing is the correct one for the organization being signed. When verifying a contract -, client -, or possibly even server certificate, the verifier can simply check whether the Provider ID matches all the way up the chain.

Using the Provider ID, third party issuance by a compromised sub-CA can be detected offline. It works well when the eMSPs and CPOs are the only parties directly under the root CA, i.e. the peer-to-peer structure explained in Section 2. However, it does not work for the centralized structure where a party can offload its certificate provisioning to a roaming hub that acts as an intermediate CA. The roaming hub is a different Provider, so this breaks the chain of identical Provider IDs up to the root.

Offline check for centralized PKI structure

To make the offline check work for the centralized structure, we could use a weakened check that only verifies that the Provider ID of the client certificate matches the Provider ID of the sub-CA certificate that signed it, but not necessarily every other certificate up to the root. However, this weakens the security guarantee from the Provider ID. Whereas in the peer-to-peer design an attacker would need to compromise an EV-charging *root* CA to defeat the check, the compromise of any roaming hub sub-CA that can issue other sub-CAs will allow it to simply issue a new sub-CA for which the Provider ID is correct. This highlights the importance of properly securing these sub-CAs.

An alternative way to make the check work for the centralized design is by issuing multiple sub-CAs to a roaming hub organization, one for each organization that it provides services to. This would allow the exact same check as in the peer-to-peer model, but it would also require additional policies such as requiring the root CA organization to verify that the roaming hub is requesting a new sub-CA for an organization that actually wants service from that roaming hub. This would be quite cumbersome, and possibly not considered worth the additional effort when considering the marginal benefit it brings over the simpler check up to the first sub-CA that signed the client certificate.

Our proposed offline check based on Provider ID should be used alongside other verification options, to solve the specific issue of offline detection of third party issuance. For the best assurance that certificates are valid, the options for online verification, such as OCSP, should also be used whenever available, because they allow instant detection of root CA or roaming hub sub-CA compromise. Online checking also allows actors to revoke individual certificates that were valid when issued, but no longer are (because e.g. the customer ended their contract).

4. THE CASE AGAINST A SEPARATE CERTIFICATE PROVISIONING SERVICE

ISO 15118 [3], the VDE application guidelines [9], and the ElaadNL guide [4] make several mentions of Certificate Provisioning. This is the process of getting contract certificates into EVs. The full details can be found in ISO 15118 [3], but to briefly summarize, Certificate Provisioning relies on a special provisioning service certificate that signs a particular message sent to the EV, giving it its new contract certificate. The EV does not need to verify the contract certificate – instead, it verifies the signature on the message.

Certificate Provisioning introduces an additional role: the party that signs the messages to carry the contract certificates is the Certificate Provisioning Service (CPS). ISO 15118 defines the CPS as a separate role which *may* be, but does not have to be, performed by the eMSP.
The reason for allowing the CPS to be a wholly separate actor introduces unnecessary security risks. The reasoning in ISO 15118 is that having a separate CPS would allow eMSPs to use sub-CAs that are not signed by an EV-charging root CA. The dedicated CPS *would* have a CPS sub-CA signed by an EV-charging root CA, and signs the messages that distribute the contract certificates to the EVs. The EV can then check at provisioning time that the certificate is in fact valid without having to know the eMSP's actual root CA.

In such a scenario, where the eMSP's sub-CA is not signed by the EV-charging root CA, offline charging cannot work. It would require the Charge Points to store additional root certificates outside of the EV-charging PKI. It multiplies the number of certificates required, and complicates key management and contract certificate provisioning. It also means that that particular eMSP's contract certificates might not be as trustworthy, because its sub-CAs and certificates may be provided by organizations who have not had to satisfy the policy requirements imposed on parties inside the EV-charging PKI. This in turn undermines the trustworthiness of the entire ecosystem.

If an eMSP wants to use an external service provider for its sub-CAs, that service provider should be required to satisfy the EV-charging PKI policies. If they do, the sub-CAs could then be cross-signed by the EV-charging root CA. Cross-signing is a mechanism where (sub-CA) certificates are signed by multiple root certificates. So the eMSP's sub-CA would have a valid path to the root certificate of its service provider, but also to the EV-charging root certificate of the EV-charging PKI. This would allow for full use of all functionality of the EV-charging PKI. It can also be viewed as simply making the external service provider part of the EV-charging PKI.

This highlights the need for *neutral* EV-charging root CAs. If the EV-charging root CA is neutral, there should be no need to use external root CAs – and ElaadNL seems to agree with that [4, pp 62].

CONCLUSIONS

The EV-charging ecosystem needs a Public Key Infrastructure (PKI). Although there are large steps towards this, there are two things we believe require some additional attention.

First, the rules for validity verification for the contract certificates need to be improved. A simple path from a client certificate to a root CA is insufficient, because of the third party issuance problem explained in Section 3. We suggest using a unique Provider ID, already part of the ISO 15118 contract certificates, to verify that certificates were issued by the organization they claim to be part of.

Second, we emphasize the importance of neutral EV-charging root CAs. If the root CAs are not neutral, there is a need for a separate Certificate Provisioning Service that allows for contract certificates not issued from within the same EV-charging PKI. This in turn means security policy enforcement is a lot harder. Conversely, if the root CA *is* neutral, there should be no need for actors to use certificates from outside the EV-charging PKI, as explained in Section 4.

As a general concern, more attention should be directed at the PKI design for EV-charging. In a separate paper [1] we analyzed several other protocols used in the EV-charging landscape. There are many different protocols (ISO 15118, OCPP, OCPI, OCHP, OICP, OSCP, OpenADR) and our analysis showed that their security guarantees are insufficient. We suggest several improvements for these protocols, such as the use of client certificates rather than static authentication credentials, and an end-to-end security mechanism such as the one published in [2]. Crucially, our suggested improvements require certificates – and thus, rely on the existence of a PKI. Several other PKIs are already in use in EV-charging, both public and private ones: some actors use the public Internet PKI to secure their server systems, others have a private PKI for client certificates in their clearing house. However, ISO 15118 is written with a *single* EV-charging PKI for the entire world in mind. It makes sense to try and consolidate the needs for *all* EV-charging protocols into that one PKI.

In its ISO 15118 PKI implementation guide ElaadNL does mention that the certificates required for other protocols in the ecosystem could be part of this PKI, but it does not define how [4]. But server- and client certificates for the other protocols fulfill different roles than the ISO 15118 contract certificates. Consolidating the PKIs currently in use into one single EV-charging PKI requires additional rules, aside from the ones established by ISO 15118. Unfortunately, most EV-charging protocols do not make their needs explicit. For example, if the protocols use TLS, they simply presuppose the existence of a PKI for TLS [1]. Additional work to determine these requirements and consolidate them into a single set of rules and policies for the envisioned EV-charging PKI is required.

REFERENCES

[1] P. Van Aubel and E. Poll, "Security of EV-charging protocols", *In submission*, arXiv:2202.04631.

[2] P. Van Aubel, E. Poll, and J. Rijneveld, Sep. 2019, "Non-Repudiation and End-to-End security for Electric-Vehicle charging", *IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pp. 1-5, doi:10.1109/ISGTEurope.2019.8905444.

[3] 2014, *Road vehicles – Vehicle-to-Grid Communication Interface – Part 2: Network and application protocol requirements*, ISO Standard 15118-2.

[4] P. Klapwijk and L. Driessen-Mutters, 2018, *Exploring the public key infrastructure for ISO 15118 in the EV charging ecosystem,* ElaadNL, Arnhem, Netherlands.

[5] N. van der Meulen, 2013, "DigiNotar: Dissecting the First Dutch Digital Disaster", *Journal of Strategic Security*, vol. 6, no. 2, pp 46-58, doi:10.5038/1944-0472.6.2.4.

[6] N. Leavitt, 2011, "Internet Security under Attack: The Undermining of Digital Certificates", *Computer*, vol. 44, no. 12, pp. 17-20, doi:10.1109/MC.2011.367.

[7] S. B. Roosa and S. Schultze, 2013, "Trust Darknet: Control and Compromise in the Internet's Certificate Authority Model", *IEEE Internet Computing*, vol. 17, no. 3, pp. 18-25, doi:10.1109/MIC.2013.27.

[8] European Court of Auditors, 2021, *Infrastructure for charging electric vehicles: more charging stations but uneven deployment makes travel across the EU complicated, Special report No 05,* Publications Office of the European Union, Luxembourg, Luxembourg, doi:10.2865/651152.

[9] Verband der Elektrotechnik Elektronik Informationstechnik e.V., 2019, *Handling of certificates for electric vehicles, charging infrastructure and backend systems within the framework of ISO 15118*, VDE Verlag GmbH, Berlin, Germany, VDE-AR-E 2801-100-1:2019-12.