Formal Verification of Privacy Properties in Electric Vehicle Charging

Marouane Fazouane[1], Henning Kopp[2], Rens W. van der Heijden[2], Daniel Le Métayer[1], and Frank Kargl[2]

[1] Inria, University of Lyon, France, marouane.fazouane@ensta.org, daniel.le-metayer@inria.fr
[2] Ulm University, Ulm, Germany, henning.kopp@uni-ulm.de, rens.vanderheijden@uni-ulm.de, frank.kargl@uni-ulm.de

Abstract. Electric vehicles are an up-and-coming technology that provides significant environmental benefits. A major challenge of these vehicles is their somewhat limited range, requiring the deployment of many charging stations. To effectively deliver electricity to vehicles and guarantee payment, a protocol was developed as part of the ISO 15118 standardization effort. A privacy-preserving variant of this protocol, POPCORN, has been proposed in recent work, claiming to provide significant privacy for the user, while maintaining functionality. In this paper, we outline our approach for the verification of privacy properties of the protocol. We provide a formal model of the expected privacy properties in the applied Pi-Calculus and use ProVerif to check them. We identify weaknesses in the protocol and suggest improvements to address them.

Keywords: privacy, formal verification, electric vehicle charging

1 Introduction

In the current practice of charging for electric vehicles, the user of the vehicle typically has a contract that allows her to charge the vehicle at any compatible charging station. The contract is comparable to a mobile phone contract, as it enables roaming by users to charging stations operated by other companies, such as an electricity provider. However, the current standards for the implementation of such contracts, which should guarantee energy delivery and payment, do not fully consider the issue of location privacy for users. For example, a charging station operator could identify users from specific energy providers, which usually operate regionally. On the other hand, the company with which the user has a contract can track her movements through the different charging stations or energy providers involved.

One of the major challenges to design a protocol for electric vehicle charging is that the privacy of the user is difficult to express. One of the sources of this complexity is the potential overlap between the responsibilities of different participants of the protocol. Broadly speaking, two different approaches have

been followed to express privacy requirements: the quantitative approach and the qualitative approach. The first category relies on the definition of appropriate "privacy metrics", such as k-anonymity, t-closeness [21], differential privacy [10, 11] or entropy [25]. The second category consists in defining privacy properties in a suitable language endowed with a formal, mathematical semantics [32], and to use verification tools to reason about these properties. This approach is akin to the traditional analysis of security protocols [5]. In this paper, we take the second approach, and study the application of formal privacy analysis to protocols, using the POPCORN electric vehicle charging protocol [14] as a case study. This protocol is based on the current draft standard [15, 16] and is thus much closer to practical application than earlier work on the privacy analysis of electric vehicle charging protocols [20]. Considering that privacy is a quite complex and multi-faceted notion, the first challenge in this context is the definition of appropriate privacy properties. In this paper, we define the privacy requirements of the protocol as a collection of six privacy properties and propose a definition of these properties in the applied Pi-Calculus [32]. Then we proceed with the analysis of the protocol and suggest several modifications to obtain an enhanced protocol meeting the required properties.

The contribution of the work presented in this paper is threefold:

- On the technical side: we provide the first realistic privacy compliant electric vehicle charging protocol improving the current draft of the ISO standard.
- On the formal side: we define privacy properties suitable for electric vehicle charging protocols including a new form of unlinkability (of the users and their uses).
- On the methodological side: we show how formal verification techniques can be applied in a "real-world" setting.

Beyond this specific protocol, this work also paves the way for a more general "privacy by re-design approach".

The remainder of the paper is structured as follows. Section 2 introduces POPCORN and discusses the relevant parts of the protocol and their relation to each other. Section 3 provides the formalization of POPCORN and discusses the relevant privacy properties. The results of our analysis are presented in Section 4, followed by suggestions of enhancements in Section 5. Related work is discussed in Section 6 and concluding remarks in Section 7.

2 Electric vehicle charging

In this paper we focus on a charging protocol for electric vehicles, which allows charging of a vehicle at a charging station in exchange for financial compensation. In practice, electricity is provided on a contract basis; therefore, financial compensation is implemented through contracts. Because energy providers often operate on a regional basis, roaming services are offered through a mobility operator. Thus, the following parties are involved in such a protocol:

Electric vehicle (EV) : This is the electric vehicle initiating the charging protocol.

Energy provider (EP) : The energy provider is the party providing the energy for recharging. This entity must receive compensation for its energy.

Charging station (CS) : The charging station is the device connecting the vehicle to the power grid to charge. In some scenarios, this may be operated by a charging station operator (CSO), although in practice this is usually the EP, who has complete control over the CS.

Mobility operator (MO) : The MO provides roaming contracts, so that the user can charge his vehicle with any EP covered by the contract. The MO has roaming agreements with one or more EPs and takes care of correct payments to these EPs as they are used by the users with whom the MO has contracts. In practice, some EPs also offer roaming contracts; thus, in some transactions, the EP and the MO may be the same entity.

This set of parties is also defined by the industry standard for electric vehicle charging, ISO 15118 [15, 16]. The protocol for charging defined by this standard is designed in such a way that no party can cheat. However it does not provide any protection against user tracking. For example, the CS can recognize and distinguish EVs based on their identifiers, which are used to authenticate the vehicle and guarantee payments. Similarly, the MO could collect a list of visited CSs to track the trajectory of an EV. These location privacy issues were not sufficiently addressed in the standard. Several of these issues were identified by Höfer et al. [14], who subsequently proposed a more privacy-preserving protocol called POPCORN.

In POPCORN, several technical and organizational measures were added to improve the privacy of the charging protocol. The following parties were added:

Dispute resolver (DR) : The dispute resolver is a trusted third party, which comes into play only if one of the parties tries to cheat (for example, by not paying, or by claiming missing payment for transactions that were already paid). The DR can then resolve the identity of the cheating party and reconstruct the necessary transactions.

Payment handler (PH) : The payment handler is a trusted third party, similar to a bank, which handles the payment process between the MO and the EP. This party should ensure that the MO learns nothing about the EP. On the other hand, the EP should also not learn from which MO he is paid. The MO knows the identities of users, but not their locations. In contrast, the EP knows locations, but is not able to link them to individuals. The PH plays the role of an intermediary between those two partial datasets.

Although POPCORN does introduce additional privacy, as argued by Höfer et al. , no formal privacy property was defined. The objective of this paper is precisely to address this issue and to challenge these privacy properties. Let us note also that the verification of integrity (non cheating) properties is not a goal of our analysis.

We now provide an outline of the different phases of the POPCORN protocol. A full specification can be found in [14].

Phase 0: Mobility contract establishment First of all, the EV user signs a contract with some MO. The EV obtains anonymous contract credentials from a global certificate authority. This could be done, e.g., with Idemix [8], which makes it possible to hide the contract credentials from the global certificate authority to ensure privacy.

Furthermore group signature credentials are installed in the EV, where the group manager is the DR. These credentials allow the EV to sign messages. Any actor can then verify that the signatures belong to a member of the group but only the DR can reveal which EV exactly provided the signature.

Phase 1: Contract authentication When an EV is plugged into a CS it establishes a link over TLS. It then proceeds to prove that its contract and its anonymous credentials have not expired. The EV does not disclose any other contract information.

Phase 2: Charging loop with meter receipts The CS delivers energy to the EV and every time a fixed quantity has been transferred, the CS sends a meter reading. The EV then has to sign this meter reading with his group signature, thus committing to the reading. The CS checks the signature and, if it is correct, the charging loop continues until the EV is fully recharged or fails to produce a correct signature.

At the end of the charging loop, the CS sends to the EV a partial service detail record. This partial SDR contains the amount of electricity the EV has received, the payable amount and the recipient of the payment, i.e., the EP. Furthermore it contains an identifier, called the transaction id, which identifies the charging session represented by a specific SDR. The information about the EP in the SDR is encrypted with the public key of the PH.

Finally, the CS anonymously sends the group-signed commitments and the partial SDR to the EP.

Phase 3 and 4: SDR delivery and payment After charging, the EV appends his probabilistically encrypted contract ID to the SDR, signs it, and forwards it to the MO. The MO can now extract the contract ID and thus the user but does not learn anything about the CS and the EP. He then sends the SDR together with the encrypted EP and the transaction number to the PH[3]. The payment handler can then recover the EP and perform the payment. The mobility operator obtains a receipt from the PH to confirm the payment.

Phase 5: Dispute resolution This phase is optional: it takes place only if the payment of the EP does not arrive withing the payment period. In this case, the EP forwards his partial SDR to the dispute resolver who can then uncover the identity of the vehicle, since he is the group manager and the SDR was signed with the private key of the EV. He then informs the MO of the missing payment and requests the payment receipt. If the mobility operator cannot provide the receipt, he has to settle the missing payment.

[3] If the EV tries to cheat here, the dispute resolution phase will allow the parties to determine the identity of the EV.

3 Formalization

In this section, we present a general framework for the formal description of cryptographic primitives and communication protocols. We then introduce privacy properties suitable for electric vehicle charging and show how to express them in this framework.

In this paper, we use the Dolev-Yao model, which is one of the standard models for analysing cryptographic protocols. The Dolev-Yao assumptions are the following: an open and unlimited network in which every agent can intercept, send and resend any message to any other agent. As a result, active adversaries can know every message exchanged in the network, and synthesize, modify, reroute, duplicate or erase messages or even impersonate any other agent. They can also use an algebraic system to derive new knowledge from the knowledge they already have. A minimal algebraic system makes it possible to create tuples and to apply projections to extract information from them. As shown in the next subsection, more powerful systems typically include cryptographic primitives and equational theories to reason about them. Even though it involves strong adversaries, the Dolev-Yao model requires additional assumptions on their computational capabilities:

- They cannot guess random numbers in sufficiently large state spaces.
- They cannot know more than what the algebraic system can prove. Typically, adversaries cannot extract the key or the plaintext from a ciphertext unless they already know the key (or the algebraic system includes rules to derive them).

Symbolic manipulations can be used to capture the properties of cryptographic primitives. Most model checkers and theorem provers used in this context abstract away from cryptography. However, results that are proven true by these tools are not necessarily sound with respect to a computational model. The computational soundness of the cryptographic primitives is needed to establish the soundness of the results. In essence, model checkers and theorem provers assume that the cryptographic primitives are secure; they only focus on proving that the *protocol* is secure, not the primitives on which it relies.

3.1 Modeling Cryptographic Primitives

The abstraction of cryptographic primitives is performed by using equational theories, which can be introduced by means of proof systems, as shown in the following. The syntax of terms is defined as follows:

$$M ::= id \mid f(M, ..., M) \mid M\{id/id\} \mid M\sigma$$

where id represents variable names and $f(M, ..., M)$ represents function application. Renaming is written $M\{id/id\}$, i.e., $f(x, y)\{z/y\}$ is equivalent to $f(x, z)$. Substitutions are written $M\sigma$ and can be used to substitute a whole term to a name: $f(x, y)\{g(t, x)/y\}$ is equivalent to $f(x,\ g(t, x))$.

We introduce two types of sequences: E is a sequence of equations over typed terms and Σ a sequence of constants and function signatures. We have the following system, which is a more natural reformulation of the equational theories discussed in [32]. The system can be easily extended with specific typing or well-formedness rules.

$$\begin{array}{rc}
\text{REFLEXIVITY} & \frac{}{E,\Sigma\vdash M=M} \\
\text{AXIOM} & \frac{(M=N)\in E}{E,\Sigma\vdash M=N} \\
\text{SYMMETRY} & \frac{E,\Sigma\vdash M=N}{E,\Sigma\vdash N=M} \\
\text{TRANSITIVITY} & \frac{E,\Sigma\vdash M=N,\ E,\Sigma\vdash N=L}{E,\Sigma\vdash M=L} \\
\text{APPLICATION} & \frac{E,\Sigma\vdash M_i=N_i,\ E,\Sigma\vdash M_i:T_i,\ i=1..k,\ E,\Sigma\vdash f:T_1\times\ldots\times T_k\to T}{E,\Sigma\vdash f(M_1,\ldots,M_k)=f(N_1,\ldots,N_k)} \\
\text{SUBSTITUTION} & \frac{E\vdash M=N}{E\vdash M\sigma=N\sigma} \\
\text{RENAMING} & \frac{E\vdash M=N}{E\vdash M\{m/n\}=N\{m/n\}}
\end{array}$$

If the theory E, Σ is decidable, we can define the congruence $=_{E,\Sigma}$ as $M =_{E,\Sigma} N \iff E, \Sigma \vdash M = N$. We can alternatively introduce this relation by means of reduction systems and we will require the system to be convergent, i.e., confluent and terminating. We will then have $M =_{E,\Sigma} N \iff \exists L, M \to^*_{E,\Sigma} L \wedge N \to^*_{E,\Sigma} L$.

Example One way to model asymmetric encryption and digital signatures, uses the following sequences:

$$\begin{aligned}
\Sigma = [&aenc : Pkey \times T \to T,\ adec : Skey \times T \to T,\ Pk : Skey \to Pkey,\\
&Sign : SKey \times T \to Signature,\ CheckSign : Pkey \times Signature \to Bool,\\
&RecoverData : Signature \to T,\ true : Bool]\\
E = [&adec(x, aenc(Pk(x), y)) \to y,\ CheckSign(Pk(x), Sign(x, y)) \to true,\\
&RecoverData(Sign(x, y)) \to y]
\end{aligned}$$

It is easy to prove that this system is convergent. For a more elaborate example, one may want to use automated tools like theorem provers together with a dedicated language to use equational theories or symbolic representation of cryptographic primitives.

3.2 Modeling Protocol Interactions and Concurrency

Process algebra provide convenient ways to formally describe high-level process interaction, communication and synchronization. They are also supported by useful tools for the formal analysis of processes. The applied Pi-Calculus is a language of this family and an extension of the well known Pi-Calculus and is

the formal language that we use to model POPCORN. It has been introduced in [32]. An Applied Pi-Calculus process is defined by the following syntax:

$$
\begin{aligned}
P ::= {} & 0 \mid phase\ n; P \mid \nu id : T;\ P \mid let\ id = M\ in\ P \\
& \mid if\ M =_{E,\Sigma} M\ then\ P\ else\ P \mid in(id, id : T);\ P \mid out(id, M);\ P \\
& \mid P|P \mid !P \\
M ::= {} & id \mid f(M, \ldots, M) \mid Const \mid \ldots \\
Const ::= {} & true \mid false \mid \ldots
\end{aligned}
$$

Informally, its semantics is defined as follows:

Null Process: 0 is the null process. It does not reduce to any other process. This is often omitted after an $in()$ or an $out()$, i.e., write $out(x, y)$ instead of $out(x, y); 0$.

Phase: $phase\ n; P$ is a synchonization point, i.e., all instructions from the previous phase are discarded before starting phase n and it behaves as P. By default, processes run in *phase* 0. In particular, this will be useful to model offline attacks.

Restriction: $\nu x : T;\ P$ creates a new typed name x, adds it to the scope of P and then behaves as P.

Let Definition: $let\ x = M\ in\ P$ binds the value of the term x to the term M; then it behaves as P.

IF-THEN-ELSE Statement: $if\ M_1 =_{E,\Sigma} M_2\ then\ P_1\ else\ P_2$ reduces to P_1 if $M_1 =_{E,\Sigma} M_2$ is provably true in the equation theoryE, Σ. It reduces to P_2 otherwise (not only when $M_1 =_{E,\Sigma} M_2$ is false, but also when it is not provable).

Input: $in(id_1, id_2 : T);\ P$ waits on channel id_1 for a term of type T to be output, reads its value into the term id_2 and then behaves like P.

Output: $out(id, M);\ P$ waits for another process to listen to the channel id (i.e. a process of the form $in(id, id_2);\ P_2$ that runs in parallel). It can then output the term M in the channel id and behaves like P.

Parallel Composition: $P_1|P_2$ basically means that both P_1 and P_2 run in parallel.

Replication: $!P$ represents an unbounded number of parallel replications of the process P.

The Pi-Calculus and the Applied Pi-Calculus have formal semantics defined by labelled transition systems. The weak bisimilarity $\approx$ is proven to coincide with the observational semantics of the language [30, 32]. This gives us a mechanical way to prove observational semantics. The latter basically means that two processes are equivalent if and only if no static context (restriction of names and parallel composition with an arbitrary process) can distinguish between the two processes. This gives us the ability to express some advanced privacy-related requirements. However, $\approx$ is undecidable and some advanced methods are needed to prove observational equivalence. For instance, ProVerif, the automated tool used in our analysis, applies specific heuristics to prove a stronger notion than observational equivalence which is decidable. For this reason, ProVerif fails to prove some valid equivalences.

3.3 Privacy-related Properties

We shall present in this subsection the main privacy properties we are interested in. We shall then show how to express and verify each of them in the Applied Pi-Calculus. The first five properties have been studied in different papers. As far as we know, this is not the case for the last property. The properties are the following:

Weak Secrecy/Confidentiality: Active adversaries cannot deduce the complete secret from their interactions with the communicating parties. Or, equivalently, adversaries cannot output the secret on a public channel [32]. This property is interesting in the context of POPCORN to express the fact that the identity of the vehicles should remain secret as well as the identities of the mobility operator (MO) and the energy provider (EP) communicating with a specific vehicle.

Strong Secrecy: This is a stronger notion where adversaries cannot even distinguish if the secret changes [32]. It provides stronger guarantees on the confidentiality of the secret, excluding any partial knowledge of the secret. In contrast, *Weak Secrecy* is not breached as long as the adversary cannot deduce the complete secret.

Anonymity: A particular user may use the service without others noticing him. This means, informally, that he can use the service without disclosing personally identifiable information during this process [7]. This property is useful in a scenario where the identity of the user (or vehicle) is known to the adversary, which can be, for example, a Charging station (CS) knowing the identity of the vehicle from a previous interaction through a non-privacy preserving protocol. In this particular scenario, the adversary should not know that a specific vehicle has used the service.

Resistance to Offline-Guessing Attacks: Adversaries cannot distinguish between correct and incorrect guesses [6].

Strong Unlinkability: A user may make multiple uses of a service without others being able to establish a link between them [7]. This property guarantees that users cannot be traced by active adversaries like CSs and EPs.

Unlinkability of uses and users: A user may use the service without others being able to link him to a particular use. This property is not to be confused with *Strong Unlinkability*. It guarantees that an active adversary, like an MO who already knows sensitive data, should not be able to link the known vehicle to a complete bill which contains charging sessions' details and metadata.

In the case of POPCORN we are interested in the first four properties for the identities of the electric vehicle (EV), its MO and the CS/EP it may communicate with. *Strong Unlinkability* is only studied with respect to the identity of the user (or EV). The last property will be studied for a particular user and a particular usage.

The above properties can be expressed as follows in the formal model:

Weak Secrecy/Confidentiality: This property can be expressed as a reachability property [32].

Strong Secrecy: The unobservability of secret changes can be captured using observational equivalence [32].
$P\{M_1/secret_1, \dots\ M_n/secret_n\} \approx P\{M'_1/secret_1, \dots\ M'_n/secret_n\}$
The above formula states that it is possible to replace secrets by different values in the protocol without active adversaries being able to distinguish these situations, which is exactly *Strong Secrecy*.

Anonymity: The fact that a particular user can remain unnoticed can be expressed as: [7]
$!(\nu id;\ P)|\ P\{M/id\} \approx !(\nu id;\ P)$

Resistance to Offline-Guessing Attacks: We can encode guesses by outputting the correct or a dummy value on a public channel. The property is true if and only of the following holds [6]:
$P|\ (phase\ 1;\ out(publicChannel,\ secret)) \approx$
$P|\ (phase\ 1;\ \nu dummy.\ out(publicChannel,\ dummy))$

Strong Unlinkability: This property amounts to checking whether the protocol is equivalent to a version of itself in which the number of sessions is limited to one per user [7].
$!(\nu id;\ !P) \approx !(\nu id;\ P)$

Unlinkability of uses and users: This property expresses the fact that is a transaction cannot be linked to a given user. Typically, in the case of POPCORN, we have two bills per charging session: one for the MO and one for the EP. If an adversary MO knows the EV and both bills, verifying the property is equivalent to answering the following question: can this adversary link a bill he knows with the bill containing the detailed charging information of the vehicle? The property is expressed as follows:
$P_1 \approx P_2$ where

- $P_1 = C[phase\ 1;\ (out(publicChannel,\ trid\_CS)|\ out(publicChannel,\ trid\_EV))]$
- $P_2 = C[phase\ 1;\ (out(publicChannel,\ dummy_1)|\ out(publicChannel,\ dummy_2))]$
- $(trid\_EV, trid\_CS)$ denotes a charging session where $trid\_EV$ is the transaction id on the user side and $trid\_CS$ is the transaction id on the charging station side.
- $dummy_1$ and $dummy_2$ are two valid but unlinked charging session identifications.
- $C[]$ an arbitrary context such that $C[0]$ represents the studied protocol.

As an illustration, a typical adversary would have the following template, where *link*() encodes strategies that can link both transactions (in ProVerif syntax):

```
let Adversary(...)=
  phase 1;
  in(publicCh, trid1:TransactID);
  in(publicCh, trid2:TransactID);
  if link(trid1, trid2, extra_infos)=true then
```

```
    (*successfully linked the usage to the user*)
    new message: MSG;
    (*'c' is public and not used elsewhere*)
    out(c,message)
```

4 Verification of Privacy Policies

The first step of our methodology consists in translating the informal description of the protocol (which may be unclear or incomplete) to simple diagrams including a complete description of each step. This representation is then translated into the Applied Pi-Calculus. The next step is the definition of the privacy properties following the approach described in the previous section. These properties are then submitted to ProVerif for verification. To this aim, we have implemented a convergent equational theory that captures all cryptographic primitives required by POPCORN. The properties that cannot be verified by ProVerif can either be shown to be incorrect or proven by hand. The failure to prove a correct property can be due either to the limitation of the tool described in Section 3.2 or because of inappropriate design choices for the model of the protocol.

4.1 Minor Problems and Adjustments

It is possible to exploit the signature of the meter readings to generate adversaries that can invalidate *Strong Unlinkability*. Indeed, a malicious CS that generates twice the same meter reading in two different sessions obtains the same signed value if and only if the EV is the same in both session (and thus iff *Strong Unlinkability* is not satisfied). We can easily confirm this claim by submitting the following equivalence to ProVerif:

```
free gmsk: gmskey [private]. (* master key *)
equivalence
( (* Multiple sessions *)
  !( new id: ID;
     !(let gsk=GKeygen(gmsk,id) in
       in(publicChannel,m:bitstring);
       out(publicChannel,(GPk(gmsk),Sign(gsk, m))) ) )
)
( (* Single session *)
  !( new id: ID;
     (let gsk=GKeygen(gmsk,id) in
       in(publicChannel,m:bitstring);
       out(publicChannel,(GPk(gmsk),Sign(gsk, m))) ) )
)
```

Automated payment gives rise to another weakness. In fact, according to the ISO/IEC 15118 specification, the EP's identification number is included in the

bill that the EV sends to the MO. In contrast, in POPCORN, an encrypted identification of the energy provider is sent to the payment handler (PH). However, the POPCORN description was imprecise about how this actually works. If a standard asymmetric encryption is used, the problem again is that even though an adversary cannot find the secret, he can still detect its changes, which means that *Strong Secrecy* is not satisfied. We can confirm this claim using the following ProVerif program, which attempts to verify observational equivalence:

```
free sk: skey [private].
free id: bitstring [private]. (* the secret *)
noninterf id. (* strong secrecy *)
process
  out(publicChannel,(Pk(sk),aenc(Pk(sk),id)))
```

Possible fixes: The above problems can be fixed as follows. First, a session number can be added to the data to be signed: $Sign(gsk,\ (session\_id,\ m))$. Signatures are now cryptographically linked to a specific session and cannot be used in two different sessions by a malicious CS/EP. For the encrypted identity of the EP an option is to use randomized encryption:

$$\nu r :\ nonce;\ aenc(Pk(sk\_PH), (r, idEP))$$

4.2 Results using ProVerif

We consider now that the changes presented in the previous subsection have been added to POPCORN. The analysis of the protocol using ProVerif returns the following results:

	true	cannot be proven
Weak Secrecy (*EV*,*MO*)	✓	
Strong Secrecy (*EV*,*MO*)	✓	
Resistance to Offline-Guessing Attacks (*EV*,*MO*)	✓	
Anonymity (*EV*)		✓
Strong Unlinkability (*EV*)		✓
Weak Secrecy (*EP*)		✓

4.3 Unlinkability of uses and users

In this subsection we show that the remaining property, *Unlinkability of uses and users*, does not hold. To do so, we will exploit two different aspects of POPCORN and prove that minor changes can lead either to a broken protocol or to a variant of POPCORN that does not verify this property. Therefore, more substantial changes are needed, which are discussed in the next section.

Exploiting automated payment Since transaction numbers are contained in the bills obtained by EPs and MOs, an adversary can simply compare the two values to link them. The linking function of the typical adversary presented in Section 3.3 would be:

```
letfun link(trid1:TransactID, trid2:TransactID)=
  trid1=trid2.
```

It is easy to verify that $P_1 \not\approx P_2$ in this case, because a message is output in channel c in the first process and not in the second one. Thus, *Unlinkability of uses and users* is not satisfied. We must find a way to generate two transaction identifiers that can only be linked by the actor generating them or by the PH. The PH should also have the ability to derive one transaction number from the other one.

Exploiting dispute resolution We consider at this point that all minor modifications suggested above have been added to POPCORN. During dispute resolution, as explained in Section 2, the EP contacts the dispute resolver (DR) with the unpaid bill. Then the latter unveils the identity of the vehicle and contacts its MO with the transaction number of the unpaid charging session. Since dispute resolution must be functional, the MO can verify that the EP-side transaction number is linked to one of the paid or unpaid sessions. The MO has the ability to link two transaction numbers. The linking function in that case would be a function corresponding to the procedure used by the MO. Thus, even with these minor modifications, *Unlinkability of uses and users* is still not satisfied. A remaining option consists in modifying dispute resolution, which is discussed in the next section.

5 Remedy

The idea behind the suggested remedy is to involve the PH in *Dispute Resolution*. To implement this solution, we need a transaction id scheme that can ensure unlinkability and unforgeability.

create(pk_PH, rand, token) : this randomized constructor returns *(transactID_user, transactID_server, pi)* such that *transactID_user* and *transactID_server* are two transaction numbers that can only be linked by the PH and pi is a Zero Knowledge Proof such that
$VerifyProof(pk\_PH', token', transactID\_server, pi) = true$ iff
$(token, pk\_PH) = (token', pk\_PH')$.

check(pk_PH,token, transactID_server, pi) returns *true* iff
$\exists rand. \exists transactID\_user.$
$(transactID\_user, transactID\_server, pi) = create(pk\_PH, rand, token)$.

getUserSideTransactionID(sk_PH,transactID_server) returns
$transactID\_user$ such that $\exists rand. \exists token. \exists pi.$
$(transactID\_user, transactID\_server, pi)$
$= create(Pk(sk\_PH), rand, token)$.

getServerSideTransactionID(sk_PH,transactID_user) returns
$transactID\_server$ such that $\exists rand.\exists token.\exists pi.$
$(transactID\_user,\ transactID\_server,\ pi)$
$= create(Pk(sk\_PH),\ rand,\ token).$

Some changes have to be made to the protocol to use the above cryptographic primitives; the corresponding diagrams can be found in Appendix A.

Transaction numbers establishment : the CS chooses the *token* and sends it to the EV through a secure channel. The latter chooses a random nonce r and uses it to generate the transaction ids: $(trid\_EV,\ trid\_CS,\ pi) = create(pk\_PH,\ r,\ token)$. The electric vehicle then sends $trid\_CS$ and pi to the charging station. The charging station checks the validity of the transaction number before using it: $check(pk\_PH,\ token,\ trid\_CS,\ pi)$.

Automated payment : upon receiving a payment order from an MO, the PH computes the id of the EP that should be contacted but also the correct transaction number that should be paid for:
$trid\_CS = getServerSideTransactionID(sk\_PH,\ trid\_EV)$. $trid\_EV$ being the transaction id on the mobility operator side.

Dispute resolution : during dispute resolution, the DR should contact the PH with an unpaid $trid\_CS$ and the identity of the MO that should be contacted. The PH will compute the correct transaction id for which the MO should pay: $trid\_EV = getUserSideTransactionID(sk\_PH,\ trid\_CS)$.

6 Related Work

The definition of appropriate frameworks to express and reason about privacy properties has generated a significant interest over the last decade. Indeed, privacy is a complex and subtle notion, and the first challenge in this area is defining formal properties that reflect the actual privacy requirements. A variety of languages and logics have been proposed to express privacy policies [2–4, 18, 17, 23, 27, 35]. These languages may target citizens, businesses or organizations. They can be used to express individual privacy policies, corporate rules or legal rules. Some of them make it possible to verify consistency properties or system compliance. For example, one may check that an individual privacy policy fits with the policy of a website, or that the website policy complies with the corporate policy. These verifications can be performed either *a priori*, on the fly, or *a posteriori*, using techniques like static verification, monitoring and audits. Similarly, process calculi like the applied Pi-Calculus have already been applied to define and verify privacy protocols [9]. Process calculi are general frameworks to model concurrent systems. They are more powerful and versatile than dedicated frameworks, which is illustrated in this work. The downside is that specifying a protocol and its expected properties is more complex. To address this issue, some authors propose to specify privacy properties at the level of architectures [1, 19]. For example, the framework introduced in [19] includes an inference system to reason about potential conflicts between confidentiality

and accountability requirements. Other approaches are based on deontic logics, e.g. [12], which focuses on expressing policies and their relation to database security or distributed systems. A difficulty with epistemic logics in this context is the problem known as "logical omniscience". Several ways to solve this difficulty have been proposed [13, 31].

Privacy metrics such as k-anonymity [22, 33], l-diversity [26] or ϵ-differential privacy [10, 11] have also been proposed as ways to measure the level of privacy provided by an algorithm. Differential privacy provides strong privacy guarantees independently of the background knowledge of the adversary. The main idea behind ϵ-differential privacy is that the presence or absence of an item in a database should not change in a significant way the probability of obtaining a certain answer for a given query. Methods [11, 28, 29] have been proposed to design algorithms meeting these privacy metrics or to verify that a system achieves a given level of privacy [34]. These contributions on privacy metrics are complementary to our work, as we follow a logical approach here, proving that a given privacy property is met (or not) by a protocol.

Liu et. al. [20] define a formal model for an electric vehicle charing protocol, differing in several ways from POPCORN. First, they do not distinguish between the MO and EP stakeholders, and they do not have a dedicated PH. Therefore they have only three parties: the user, the supplier, which in POPCORN is called the EV, and the judging authority, which is comparable to DR in POPCORN. Their protocol [24] also supports additional functionalities such as traceability (if the car is stolen), which is not proven to be privacy preserving and discharging of the EV, i.e., the EV can choose to sell energy back into the grid.

7 Conclusions

This paper presents an application of a formal approach to define a real life protocol meeting privacy requirements. Our formal model has made it possible to identify weaknesses in the original POPCORN protocol [14] and to suggest improvements to address these issues. POPCORN preserves the confidentiality of its users (*Weak Secrecy*) but *Strong Secrecy* and *Strong Unlinkability* are not satisfied by the original version of the protocol. However, minor modifications of the protocol are sufficient to redress these weaknesses. We have also shown that POPCORN does not ensure a particular form of unlinkability: it does not prevent an attacker from linking a user to his uses of the system. We have also argued that more significant changes in the definition of POPCORN are necessary to address this issue. The mitigation proposed here does not affect the functionality of the protocol and can be shown to meet the expected unlinkability property.

The work described in this paper can be seen as a contribution to privacy re-engineering which is of prime importance to enhance legacy systems to deal with privacy requirements. The next step in this direction would be to go beyond this specific protocol and provide a framework for privacy re-engineering. We believe that the re-design approach presented in [14] in association with the formal approach described here pave the way for the definition of an iterative improve-

ment methodology that could form the core for such a framework. We would also like to stress that this approach should not be opposed to the "privacy by design" philosophy. Indeed, privacy requirements very often are (or seem to be) in conflict with other (functional or non functional) requirements. The iterative methodology suggested here could be applied at the level of specifications and seen as a strategy to address the needs on privacy by design in some situations.

References

1. Antignac, T., Le Métayer, D.: Privacy by Design: From Technologies to Architectures. In: Privacy Technologies and Policy, LNCS, vol. 8450, pp. 1–17. Springer (2014)
2. Backes, M., Dürmuth, M., Karjoth, G.: Unification in Privacy Policy Evaluation - Translating EPAL into Prolog. In: POLICY. pp. 185–188 (2004)
3. Barth, A., Mitchell, J.C., Datta, A., Sundaram, S.: Privacy and Utility in Business Processes. In: CSF. pp. 279–294 (2007)
4. Becker, M.Y., Malkis, A., Bussard, L.: A Practical Generic Privacy Language. In: ICISS. pp. 125–139 (2010)
5. Blanchet, B., Abadi, M., Fournet, C.: Automated verification of selected equivalences for security protocols. In: Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on. pp. 331–340. IEEE (2005)
6. Blanchet, B., Smyth, B.: Proverif 1.85: Automatic cryptographic protocol verifier, user manual and tutorial (2011)
7. Brusó, M., Chatzikokolakis, K., Etalle, S., den Hartog, J.: Linking Unlinkability. In: Palamidessi, C., Ryan, M. (eds.) Trustworthy Global Computing, Lecture Notes in Computer Science, vol. 8191, pp. 129–144. Springer Berlin Heidelberg (2013)
8. Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: Proceedings of the 9th ACM conference on Computer and communications security. pp. 21–30. ACM (2002)
9. Delaune, S., Kremer, S., Ryan, M.D.: Verifying Privacy-type Properties of Electronic Voting Protocols. Journal of Computer Security 17(4), 435–487 (Jul 2009), http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DKR-jcs08.pdf
10. Dwork, C.: Differential Privacy. In: ICALP (2). pp. 1–12 (2006)
11. Dwork, C.: A firm foundation for private data analysis. Commun. ACM 54(1), 86–95 (2011)
12. Glasgow, J., MacEwen, G., Panangaden, P.: A logic for reasoning about security. In: Proc. of the 3rd Computer Security Foundations Workshop. pp. 2–13 (1990)
13. Halpern, J.Y., Pucella, R.: Dealing with Logical Omniscience. In: Proc. of the 11th Conf. on Th. Aspects of Rationality and Knowl. pp. 169–176. ACM, USA (2007), http://doi.acm.org/10.1145/1324249.1324273
14. 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. ACM (2013)
15. ISO: Road vehicles – Vehicle-to-Grid Communication Interface – Part 1: General information and use-case definition. ISO 15118, International Organization for Standardization, Geneva, Switzerland (2012)
16. ISO: Road vehicles – Vehicle-to-Grid Communication Interface – Part 2: Technical protocol description and Open Systems Interconnections (OSI) layer requirements. ISO 15118, International Organization for Standardization, Geneva, Switzerland (2012)

17. Jafari, M., Fong, P.W.L., Safavi-Naini, R., Barker, K., Sheppard, N.P.: Towards defining semantic foundations for purpose-based privacy policies. In: CODASPY. pp. 213–224 (2011)
18. Le Métayer, D.: A Formal Privacy Management Framework. In: FAST (Formal Aspects of Security and Trust). pp. 161–176. Springer, LNCS 5491 (2009)
19. Le Métayer, D.: Privacy by Design: A Formal Framework for the Analysis of Architectural Choices. In: Proc. of the 3rd ACM Conference on Data and Application Security and Privacy. pp. 95–104. ACM, USA (2013), http://doi.acm.org/10.1145/2435349.2435361
20. Li, L., Pang, J., Liu, Y., Sun, J., Dong, J.S.: Symbolic analysis of an electric vehicle charging protocol. In: Proc. 19th IEEE Conference on Engineering of Complex Computer Systems (ICECCS'14). IEEE Computer Society (2014)
21. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In: IEEE 23rd International Conference on Data Engineering. pp. 106–115 (Apr 2007)
22. Li, N., Qardaji, W.H., Su, D.: Provably Private Data Anonymization: Or, k-Anonymity Meets Differential Privacy. CoRR abs/1101.2604 (2011)
23. Li, N., Yu, T., Antón, A.I.: A semantics based approach to privacy languages. Comput. Syst. Sci. Eng. 21(5) (2006)
24. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Enhancing location privacy for electric vehicles (at the right time). In: Computer Security–ESORICS 2012, pp. 397–414. Springer (2012)
25. Ma, Z., Kargl, F., Weber, M.: A location privacy metric for V2X communication systems. In: IEEE Sarnoff Symposium, 2009. pp. 1–6 (Mar 2009)
26. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: Privacy Beyond k-Anonymity. In: ICDE. p. 24 (2006)
27. May, M.J., Gunter, C.A., Lee, I.: Privacy APIs: Access Control Techniques to Analyze and Verify Legal Privacy Policies. In: CSFW. pp. 85–97 (2006)
28. McSherry, F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun. ACM 53(9), 89–97 (2010)
29. McSherry, F., Talwar, K.: Mechanism Design via Differential Privacy. In: FOCS. pp. 94–103 (2007)
30. Milner, R.: Communicating and Mobile Systems: The Pi-calculus. Cambridge University Press, New York, NY, USA (1999)
31. Pucella, R.: Deductive Algorithmic Knowledge. CoRR cs.AI/0405038 (2004)
32. Ryan, M.D., Smyth, B.: Applied pi calculus. In: Cortier, V., Kremer, S. (eds.) Formal Models and Techniques for Analyzing Security Protocols, chap. 6. IOS Press (2011), http://www.bensmyth.com/files/Smyth10-applied-pi-calculus.pdf
33. Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 557–570 (2002)
34. Tschantz, M.C., Kaynar, D.K., Datta, A.: Formal Verification of Differential Privacy for Interactive Systems. CoRR abs/1101.2819 (2011)
35. Yu, T., Li, N., Antón, A.I.: A formal semantics for P3P. In: SWS. pp. 1–8 (2004)

A POPCORN v2

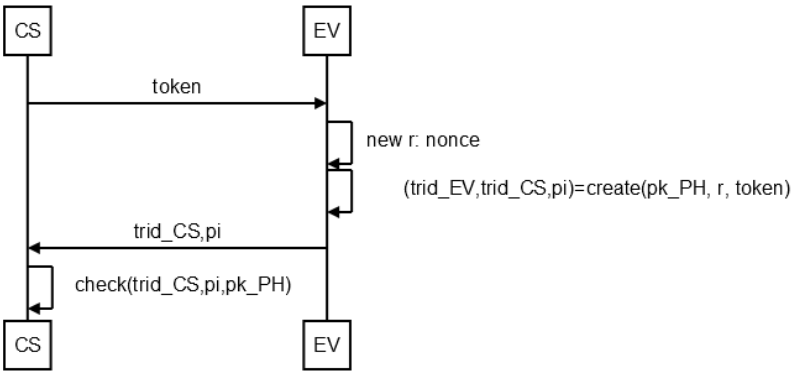

Fig. 1. Transaction number establishment

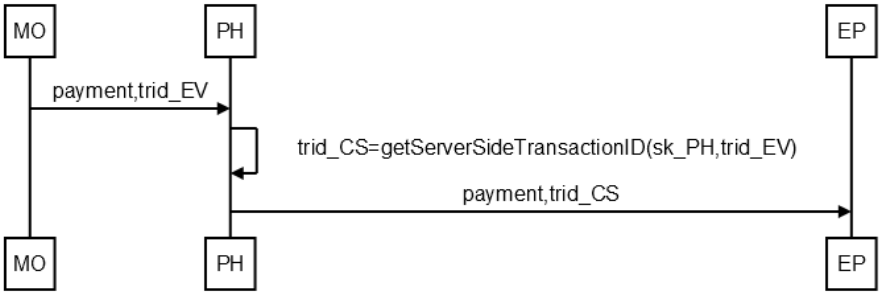

Fig. 2. Changes in automated payment

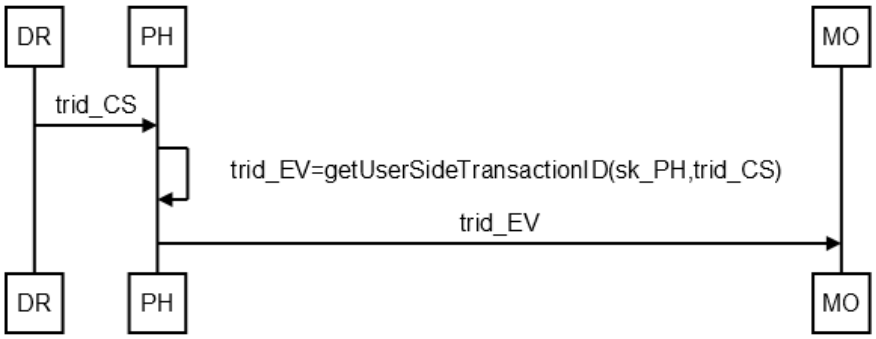

Fig. 3. Changes in dispute resolution