Provably Privacy-Preserving Distributed Data Aggregation in Smart Grids

Marius Stübs, Tobias Mueller, Kai Bavendiek, Manuel Loesch, Sibylle Schupp, Hannes Federrath

Provably Privacy-Preserving Distributed Data Aggregation in Smart Grids

Marius Stübs[1], Tobias Mueller[1], Kai Bavendiek[2], Manuel Loesch[3], Sibylle Schupp[2], and Hannes Federrath[1]

[1] University of Hamburg
{stuebs,mueller,federrath}@informatik.uni-hamburg.de
[2] Hamburg University of Technology
{kai.bavendiek,schupp}@tuhh.de
[3] FZI Research Center for Information Technology, Karlsruhe
loesch@fzi.de

Abstract. The digitalization of power systems leads to a significant increase of energy consumers and generators with communication capabilities. Using data of such devices allows for a more efficient grid operation, e.g., by improving the balancing of power demand and supply. Fog Computing is a paradigm that enables efficient aggregation and processing of the measurements provided by energy consumers and generators. However, the introduction of these techniques is hindered by missing trust in the data protection, especially for personal-related data such as electric consumption. To resolve this conflict, we propose a privacy-preserving concept for the hierarchical aggregation of distributed data based on additive secret-sharing. To increase the trust towards the system, we model the concept and provide a formal proof of its confidentiality properties. We discuss the attacker models of colluding and non-colluding adversaries on the data flow and show how our scheme mitigates these attacks.

Keywords: smart grid security · smart metering · formal model · automated proof · additive secret sharing · distributed and decentralized security.

1 Introduction

The electricity consumption of private households is usually not monitored nor managed in real-time. Traditionally, the power meter aggregates the electricity flow over the year and the aggregated value is only read for billing. However, the continuing increase in the share of renewable energies will require small-scale consumers and producers to actively participate in the demand-supply matching process. In order to gain more insight in the power flows within their grids, grid operators have strong interests in obtaining additional measurements from their customers. Power suppliers also want to allow their customers to benefit from price fluctuations at wholesale electricity markets. The roll-out of smart meters

sets the foundation for Smart Grid services such as selective data aggregation and distributed power balancing [48]. For the grid operators, an important use case is power system state estimation, where measurements from different sources are aggregated to provide a more precise overview of the respective distribution grid.

In the context of Smart Grid, privacy is a major concern for consumers and prosumers [37]. Depending on the precision of the measurement data, researchers were even able to distinguish between TV channels and thereby identify the movie that was being displayed on a specific type of home TV screens [1]. This can also reveal other personal information and daily routines, such as how many inhabitants are home and when they leave or return. In Europe, the legislator demands data to be processed in ways that have been designed to respect the privacy of the users ("Privacy by Design and Default") [24, §25]. Whether households agree to participate in local market schemes or choose to demand cloud-based Smart Grid services therefore heavily depends on the perception of these services as serious and trustworthy. One way to increase the plausibility and transparency of cloud-based applications is to incorporate security measures already in an early stage and explain as well as verify these measures. This is where formal modeling and automated proofs come into play to back the security-related claims of the service providers.

1.1 Fog Computing in Power Grids

In power grids, the control of consumers and generators can be hierarchically aggregated for provisioning of smart grid services at different grid levels. Smart grid services are, e.g., power adjustments required in the demand-supply matching process. Technically, they are realized by Energy Management Systems (EMSs). The aggregation of control options and data allows for improved decisions as further information such as the grid structure can be considered at higher aggregation levels.

At the lowest aggregation level, Nano Grids can be recognized. Examples for Nano Grids are buildings with an EMS that locally processes data and balances power demand and supply within the building. They also may provide grid services and data to external Smart Grid Service Providers or higher-level aggregators. Nano Grids are always dependent on the connection to the main grid [39]. At a higher aggregation level Micro Grids can be recognized. Micro Grids are often defined to be self-sufficient in the sense that they can support islanding and that they, in case of emergency, can encapsulate themselves from the higher-level distribution grid. Examples for Micro Grids are districts with an EMS that processes data of multiple buildings for balancing the district's power demand and supply. Control options and data provided by multiple Micro Grids can be aggregated by an EMS at the level of the corresponding distribution grid which is operated by a Distribution System Operator (DSO). Finally, control options and data of multiple distribution grids can be aggregated by an EMS on the level of the transmission grid which is operated by a Transmission System Operator (TSO). EMSs at different aggregation levels allow for different smart

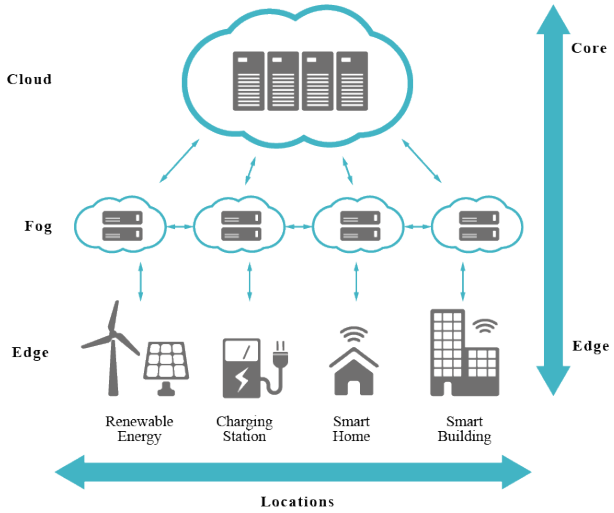

Fig. 1. The edge computing schematic.

grid services. In particular DSOs and TSOs are highly regulated regarding the smart grid services they have to provide. DSOs are responsible for voltage control and TSOs are responsible for frequency control.

The hierarchical structure of Fog Computing is visualized in fig. 1. It fits well to the structure of electric grids, especially when extending the fog node layer to a multi-layer architecture. In this paper we propose to extend previous approaches that aggregate values in Fog nodes and to implement a hierarchical aggregation scheme for sensor data that resembles the structure of the power lines. Aggregated sensor data (such as aggregated power values of multiple districts within a distribution grid) set the foundation for the realization of Smart Grid services that are ensuring grid stability.

1.2 Smart Metering & Data Security

In order to facilitate the integration of fluctuating renewable energy resources and to improve the demand-supply matching, the European Union directive 2009/72/EG requires member states to install smart metering infrastructures. As a consequence, it can be expected that a large share of households will deploy EMSs in the near future [48]. In this context, Smart Meter Gateways (SMGWs) provide a communication link between Nano Grids such as Smart Buildings and external parties. This communication link offers access to smart meter data and hence sets the foundation for the provisioning of grid services to Smart Grid Service Providers or higher-level aggregators.

The aggregation of data is a security critical function that needs to maintain the users' privacy by keeping the data as confidential as possible. Data about electricity consumption contains information about the user's habits. The more

fine-grained the data, the better can the attacker infer details about the subject. Not only has it been shown that it is possible to detect the appliances a consumer is using [33], it is also possible to infer what TV program the consumer is watching [25]. The aggregation of data must thus be private in that the parties involved learn as little as possible.

1.3 Data Flow Modeling and Automated Proofs

Several approaches exist to show that a system works as intended. Testing is a well known technique, which, however, cannot demonstrate the absence but only the presence of errors. Formal methods on the other hand follow another approach by formalizing the system in way such that certain properties can be proven. One example of a formal method is model checking where a formal model is verified to follow a certain specification. The specification can be expressed using formal properties. To actually verify formal properties of a system, one has to make a formal description of the system. The complexity of real-life systems might be neither beneficial nor necessary for the verification of certain properties. Therefore, a formal model can be a good abstraction of, for instance, the data flow of a system.

1.4 Structure of the Paper

Section 2 comprises a comparative review of related work. In section 3 the fundamental design decisions and goals are elaborated. The general data flow privacy validation scheme is described in section 4. The concept of hierarchical data aggregation is elaborated in section 5 and is then evaluated in the security discussion in section 6. We conclude the paper with a short summary in section 7.

2 Related Work

The proposed concept combines two areas of research, namely Fog Computing based privacy-preserving data aggregation and modeling of privacy-respecting data flows. In this section, we present the work related to each field.

The transition to smart metering and the Smart Grid evoked a lively scientific discussion about the impacts on privacy the transition entails [41,47,5]. With the discussion, several approaches, mitigations, and solutions have been proposed ranging from adding noise to the actual usage pattern to make analysis harder [31] to privacy-preserving data aggregation techniques to be used in Smart Grids [20,19]. A different approach on private data protection is using batteries to add noise to hide usage pattern [31]. Over the course of the last decade, the use of secret-sharing, additive or otherwise, for aggregating smart meter readings has been proposed [35,12,15], as have other privacy preserving mechanisms [21,29], such as ensuring ϵ-differential private aggregation [8,18].

Due to the recent interest on Cloud and Fog Computing, several publications (e.g. [42,44,43,4,16]) that apply cloud-based solutions to deal with the

data explosion challenge in a Smart Grid. Especially when it comes to reducing the communication bandwidth between the smart meter and cloud in cloud computing, regional [45] or hierarchical [46] aggregation schemes based on Fog Computing are still not fully researched. Moreover, data security and privacy are also critical issues when sensitive smart meter data is aggregated in only partially trusted environments such as Fog nodes and cloud applications.

In the field of privacy-respecting data flow verification different approaches exist to model data flow and verify privacy properties. A survey paper by Gürses, Troncoso, and Diaz shows different privacy properties and case studies including homomorphic encryption in an automated toll pricing system [26]. A Smart Grid case study with focus on smart metering is described in [17]. It has the same research question but makes a distinction between accountable and private readings. Application-specific approaches range from e-government [28,30] over e-healthcare [36,32] and medical registers [10] to cloud computing [11,49]. Approaches based on the applied pi-calculus [34,6,13] are popular for protocol-oriented applications with focus on data integrity (often in e-voting systems). ProVerif is one of the few tools in this field, which implements a typed version of the applied pi-calculus. Another category of formal methods are approaches based on type systems [23,38,22]. These papers usually aim at achieving differential privacy by using typing rules. Other approaches, like this paper, base their modeling on software architectures. Some of these employ modal logics like modified epistemic logic [2,3,27,9] or temporal logic for reasoning about data minimization [7]. Of the former ones CAPRIV [2] and CAPVerDE [9] are tools that support the modeling and verification of data minimization properties in software architectures. However, to the best of our knowledge, the tool CAPRIV and its source are not openly accessible. The open source tool CAPVerDE is a very similar tool that operates on software architectures and a modified epistemic logic with focus on data minimization and access control.

3 Models and Design Goals

We consider the electricity supply and demand in a Micro Grid. We assume that the Micro Grid is properly designed such that a portion of the electricity demand related to basic living usage (e.g., lighting) from the residents, termed basic usage, can be guaranteed by the minimum capacity of the Micro Grid. There is randomness in both electricity supply (due to, e.g., weather change) and demand (e.g., entertainment usage in weekends). To cope with the randomness, the Micro Grid works in the grid-connected mode and is equipped with energy storage systems (ESSs), such as an electrochemical battery, superconducting magnetic energy storage, flywheel energy storage, etc. The ESSs store excess electricity for future use.

3.1 Modeling of Privacy-Respecting Data Flows

For this paper we make use of a logic and tool called *CAPVerDE* [9]. The logic consists of a formal description language for software architectures, a data flow

property language, and a rule-based verification system. With these building blocks we can describe the data flow of a software system, express properties like "Actor A should have access to Data d", and verify whether the specified model satisfies said properties. The tool aids us by automatically solving the inference-rule based satisfaction problem.

The process of using this technique is to first identify the relevant data flow, actors, and data dependencies of the system. Then use a formal description language to describe the data flow. Setting up a formal model is already a big step because one decides which details are important and where to abstract. Once the system is modeled, one has to formalize the design goals for the system. This is done using a logic that allows for certain statements about the model. The verification rules allow for checking whether such a property holds for the provided model. If the verification returns that the model satisfies all properties, the model meets the design goals.

For the specific use case of this paper, we want to look at privacy design goals. These are usually expressed by properties that state that an 'untrustworthy' actor should **not** have access to data that is considered personal or worth of protection. For a Smart Grid architecture this means, for instance, protecting measurements of smart meters against curious intermediate aggregators. To verify such a property, we have to model the data flow from smart meters to service providers, including the implicit data flow, i.e., dependencies between data.

In order to achieve the goals that we have modeled, we make use of an additive secret-sharing-based multi-party computation as suggested in the past [35,12,15]. With that technique, we can achieve the private aggregation of values and prevent disclosure of single measurements. However, we are also interested in hiding the aggregated values from unauthorized intermediary nodes, so we additionally share a secret with the intended receiver of the aggregated result. With this approach we can have the values aggregated by any number of nodes in the Fog, but only the intended recipient can remove the blinding and obtain the actual result.

3.2 Network Model

The roles of the proposed encryption scheme are depicted in fig. 2. Actors in the distribution grid are a) Smart Meter Gateways (SMGWs) that collect and disseminate measurement data, b) Intelligent Energy Devices (IEDs) that are controllable via a interface and consume electricity or in case of distributed energy resources inject power into the distribution grid, c) data aggregators that operate on Fog nodes to collect, aggregate and redirect measurement data, and d) Smart Grid Service Providers who receive the aggregated measurements. The e) key management authority enables confidential end-to-end connections between the other actors.

a) Smart Meter Gateway (SMGW): The measurements from the IEDs are encrypted and disseminated by the SMGW. The data is encrypted using secret-sharing, so that is can be aggregated without decryption.

b) Intelligent Energy Devices (IED): For our scheme, the readings from the

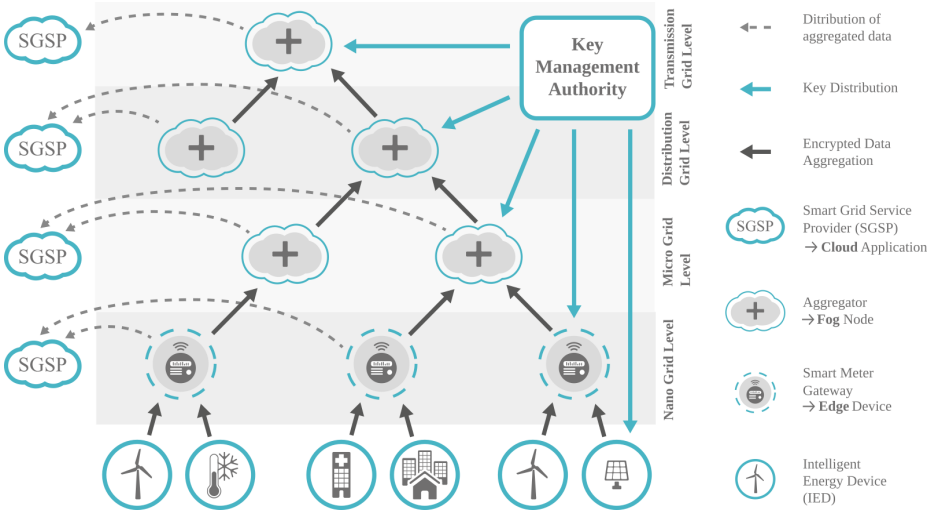

Fig. 2. The lowest layer includes the consumers and generators. In the higher layers, only aggregators reside. Aggregators can not decrypt the aggregated values. Possible data flow is indicated by the arrows. Only the Smart Grid Service Providers have access to the aggregated data. Authorization is based on the aggregation level.

IED are communicated to local SMGWs. The produced data is then encrypted by the SMGW and disseminated to the correct data aggregators.
c) Data Aggregators (DA): The DAs receive encrypted data that are then aggregated for the use either by authorized Smart Grid Service Provider, or by the next level data aggregators.
d) Smart Grid Service Providers (SGSP): Each Smart Grid Service Provider is associated with an aggregation layer. The authorization to read the aggregated measurements is expressed in a "layer key" which allows them to decrypt respective data sets.
e) Key Management Authority (KMA) The KMA provides a central trust infrastructure for the data flow. To establish a trusted link between the SMGWs, the DAs and the Smart Grid Service Providers, the KMA distributes secrets for each hierarchical aggregation layer.

3.3 Threat Model

The presented scheme is designed to provide the confidentiality of data with respect to hierarchical layers in the aggregation architecture. Following the formal modeling approach, we can distinguish between two types of adversaries: Colluding and non-colluding attackers.

Non-Colluding Attackers For the non-colluding attacker, we consider the honest-but-curious attacker model [14], where the respective node complies to

the protocol in such that they are only collecting and analyzing the data that they lawfully receive. This type of adversary does not forget any data once received. They try to computationally and statistically break any obfuscation on the received data to break confidentiality where possible. The attacker model is called honest-but-curious because they do not actively request data if not obliged to by the protocol.

For the evaluation, we consider all individual actors – that is edge nodes, Fog nodes and cloud services – as potential attackers and formally model the requirement that the proposed scheme has to protect the personal data from unauthorized access.

Colluding Attackers In extension of the previous model, we also formally describe that nodes communicate their knowledge among each other to collaborate on breaking the confidentiality. The colluding attacker actively shares information with other nodes and combines this acquired knowledge. Manipulation of data, selective forwarding and forging attacks are not considered because the focus of this paper is on privacy and data flow. However, we consider out-of-band forwarding (via gossiping) of confidential communication to colluding nodes. We assume an attack to be successful, if two (or more) colluding nodes can derive knowledge that exceeds the knowledge from both (all) respective nodes individually.

4 Formal Privacy Modeling Methodology

This section presents the formal method to model and analyze the Smart Grid system under investigation and its desired privacy properties.

4.1 Data Flow Description Language

Table 1 shows a relevant excerpt of the syntax of the architecture description language of CAPVerDE. An architecture A consists of so-called relations R that model explicit and implicit data flows between the actors (called components C_i). The relation $Has_i(X)$ is the entry point, i.e., it models how data enters the system. The relations can be read as "Component C_i has access to a variable X" for example via a sensor. $Rec_{i,j}(\{X\})$ is the relation that models the passing of data from one actor to another. It can be read as "Component C_i receives the set of variable $\{X\}$ from component C_j." $Comp_i(X = T)$ is the relation that lets actors derive new data from already known data. The natural language description of this relation is "The Component C_i can compute variable X from some term T", where term T consists of already known variables. The language also allows for implicit data flow, that is, calculations that are not intended in the system but possible. Relation $Dep_i(Y, \{X^1, ..., X^n\})$ states that "Component C_i has the computational power to derive variable Y from the set of variables X_i." For example if it is known that $a = b + c$, all actors in the system will have the relation $Dep_i(a, \{b, c\})$. The absence of such a relation denotes the

fact that a calculation is not feasible. The addition is not directly reversible, i.e., one cannot derive the summands from the sum alone, therefore the relation $Dep_i(b, \{a\})$ will not exist. It should be noted that, although this type of relation deals with implicit data flow, the relations have to be modeled explicitly, that is, for each equation one has to specify whether it should be reversible. Also, all dependencies of a variable have to be made explicit in this way, as well.

Table 1. Syntax of architecture language

A	$::= \{R\}$	
R	$::= Has_i(X)$	$\mid Rec_{i,j}(\{X\})$
	$\mid Comp_i(X = T)$	$\mid Dep_i(Y, \{\ X^1, ..., X^n\})$

A formal software system is defined as a set of relations. To make the description more readable for humans, we will provide a graphical representation of the system. The graphical form shows the components as boxes containing the corresponding relations and arrows between component boxes show the inter-component relations.

The verification process of CAPVerDE works as follows: The modeling of the architecture has to be done by the user. One follows the syntax of the language to describe the data flow. Also, the dependencies between the data has to be expressed. CAPVerDE provides a GUI to aid the modeling process. The tool then automatically checks the architecture for consistency, i.e., whether the data flow is possible. For instance, receiving data from a component that does not possess the data would be inconsistent. If the architecture is consistent, the privacy goals have to be expressed as properties in the provided language. This can also be done using the GUI. The verification of selected properties is done automatically by the tool. It returns "property holds" or "property does not hold" for the given architecture. One can view the verification trace of a property to inspect the applied rules necessary for the verification.

4.2 Privacy Goal Formalization

The only type of property we consider for this paper is the one regarding the access of data. A property $\phi = HasAcc_i(X)$ states that component C_i is able to access variable X. An architecture satisfies this property (denoted by $A \vdash \phi$) iff the component can use its explicit and implicit data flow to derive X. Also negation of properties is possible. A negated property $\neg\phi$ holds iff the property ϕ does not hold.

Table 2 shows the rules of inference that are used by the automatic verification process. The first three rules (**H1-H3**) regard the explicit data-flow and

Table 2. Rules of inference for the architecture logic in CAPVerDE

$$\textbf{H1}\ \frac{Has_i(X) \in A}{A \vdash HasAcc_i(X)} \qquad \textbf{H2}\ \frac{Rec_{i,j}(E) \in A \quad X \in \{E\}}{A \vdash HasAcc_i(X)} \qquad \textbf{H3}\ \frac{Comp_i(X = T) \in A}{A \vdash HasAcc_i(X)}$$

$$\textbf{H4}\ \frac{Dep_i(X, \{X^1, ..., X^n\}) \quad \forall l \in [1, n], A \vdash HasAcc_i(X^l)}{A \vdash HasAcc_i(X)}$$

hold if the component somehow processes the data. The fourth rule (**H4**) is based on the dependence relation and hence on the computational ability of the component to derive the data.

5 Privacy-Preserving Aggregation

This section presents the integrated concept for managing the aggregation of data in Fog Computing based Smart Grid networks. We first introduce the prerequisites for our scheme, namely a central trusted party which manages the participants' keys. Then we describe the enrollment phase of devices in the network, before we discuss the actual steps the nodes in the network have to perform. We present two protocols: One for disguising the individual readings of a smart meter and another for blinding even the aggregate value to unauthorized aggregators.

Our scheme follows the privacy enhancing architecture of smart metering proposed by Molina-Markham et al [40]. More precisely, we use zero knowledge protocols for communication assume secure communication between the utility service provider and the nodes in the Smart Grid topology. Additionally, we make use of a secret-sharing-based approach as has been proposed multiple times in the past [35,12,15].

Our scheme consists of two procedures. The first procedure, explained in section 5.1, is to hide individual readings from the aggregators. The second procedure is to hide the actual aggregation result from the aggregators themselves, such that only the final recipients, the respective Smart Grid Service Providers, can decrypt the value. This second procedure is described in section 5.2

5.1 Procedure 1 – Hiding individual readings from the aggregators

We refer to fig. 2 for a description of the actors. Let each of the n home-level IEDs be IED_n. Each IED_j creates secret-shares of its reading m^j: $m^j = \sum_{i=0}^{n} m_i^j$ and disseminates each share m_i^j to IED_i where $i \neq j$. Each IED_j then accumulates all the shares they have received $x_j = \sum_{i=0}^{n} m_i^j$ and sends the result on to the data aggregator DA. The DA can calculate $\sum_{j=0}^{n} x^j$. This is equal to the accumulated readings: $\sum_{j=0}^{n} m^j$.

This procedure is elegant, because it works locally without a centralized key management authority. It protects the confidentiality of all participating child nodes resp. their contribution. On the other hand, it does not protect the confidentiality of the sum since the aggregator learns the aggregated value. Also, if all child nodes but one collude, they can unveil the measurements of the remaining node. Therefore, we improve the scheme in order to hide both the sum and all the participating nodes.

5.2 Procedure 2 – Hiding the sum from the aggregators

We extend the previous protocol by further blinding the values with a secret-share from the intended receiver of the aggregated values. That is, we keep the structure of the participants, but rely on a central key management authority (cf. section 3.2) to generate and distribute keys.

First, the key management authority (KMA) generates a key k and generates shares for each of the n $IEDs$: $k = \sum_{j=0}^{n} k_j$. Those shares are then sent to each IED_j, which then re-shares that key: $k_j = \sum_{i=0}^{n} k_j^i$ where $i \neq j$. Each IED_j then blinds the shares of its reading (m_i^j) with the re-shared key, before sending it on to the other $IEDs$: $m_i^j + k_i^j$. Each IED_j accumulates their received shares: $x_j = \sum_{i=0}^{n} m_i^j + k_i^j$ and sends the result on to the aggregator DA, who then calculates the sum as $\sum_{j=0}^{n} x^j$. This is equal to the accumulated readings with the blinding: $\sum_{j=0}^{n} m^j + k_j = k + \sum_{j=0}^{n} m^j$. Now, only the holder of k can remove the blinding and thus obtain the actual aggregate.

6 Proof of Privacy-Preserving Data Dissemination

In this section we describe the actual proof of privacy preservingness. The high-level security threats as described in section 3.3 are examined and it is discussed how data confidentiality is achieved.

6.1 Procedure 1

Figure 3 shows the architecture of Procedure 1 (cf. section 5.1). Three aggregators for the street-level and one for the district-level are considered. Also for the sake of readability only the rightmost components of each level report to a service provider. The variables m_a, m_b, m_c are measurements of the Intelligent Energy Devices and $value0_a, value0_b, value0_c$ are the aggregated measurements on the smart meter level. $value1_a, value1_b, value1_c$ are the aggregated measurements of the street level and $value2$ is the aggregated measurement of the district level. All other variables are intermediate results and shared secrets between the components of one level. $c0_1$ ad $c0_2$, for instance, are the shared secrets of $SMGW_c$, which are distributed between $SMGW_a$ and $SMGW_b$.

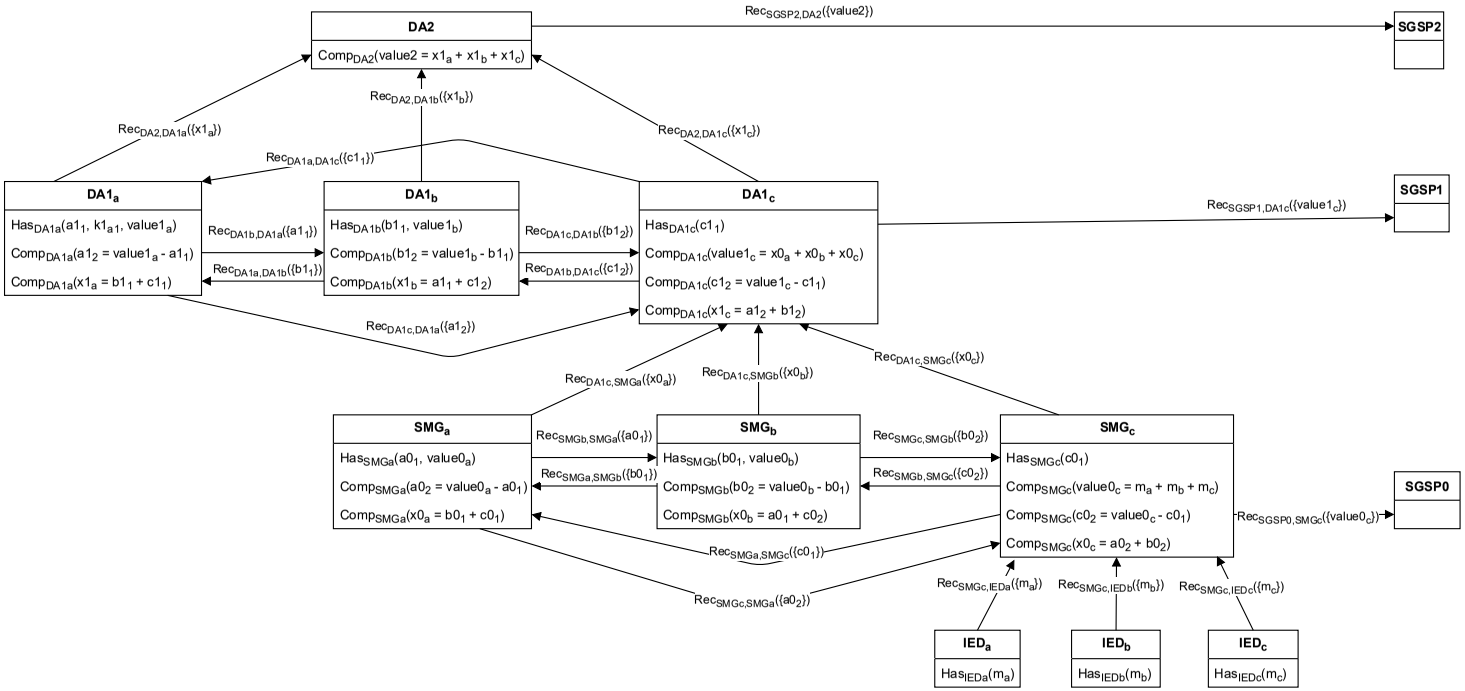

Fig. 3. Graphical representation of the Procedure 1 in the formal description language of CAPVerDE.

The depicted graph describes the explicit data flow of the architecture. As mentioned above, the implicit data flow is described via the dependence relations. For this architecture we assume that all components have the computational power to build sums and differences. Therefore, all components have the same set of dependence relations. All components can perform the explicit calculations done by other components on the system as well as all corresponding equivalent calculations. This is based on the assumption that the protocol is known. For example the set of dependence relations of SMGW A includes the dependence relations $dep_{SMGWa}(c0_2, \{value0_c, c0_1\})$ and also the derived $dep_{SMGWa}(value0_c, \{c0_1, c0_2\})$ and $dep_{SMGWa}(c0_1, \{value0_c, c0_2\})$.

Non-Colluding Attackers In this setting we want to verify that no single actor can access measurements that are not intended for them. To to do so we can verify that components on the same level cannot access each others' values and components on a higher level cannot read individual readings from the lower level. An example of a property of the first type in the syntax of CAPVerDE is $\phi_{1n} = \neg HasAcc_{SMGWa}(value0_c)$. The SMGW A should not be able to read the measurement of SMGW C (same level). An example of a property of the second type is $\phi_{2n} = \neg HasAcc_{AG1c}(value1_c)$. The Aggregator C (street level) should not be able to read the summed measurements of the SMGWs. We refer to appendix A for the discussion of the verification trace.

Colluding Attackers Here we consider colluding attackers in the model, that is two components sharing their knowledge to obtain data that both cannot access individually. To model such behavior we have to manually add "attack-relations" to the architecture. To stay with the existing example, we let the SMGWs A and B collude to access the readings of C. Therefore, we add the relation $Rec_{SMGWa,SMGWb}(\{c0_2\})$ to the architecture to model that B shares its secret with A.

The property $\phi_{1c} = \neg HasAcc_{SMGWa}(value0_c)$ fails for the architecture with the additional collusion-relation added. The verification trace looks identical to the one for the non-colluding attacker (depicted in fig. 5) up to line 37. Rule **H2** for property $HasAcc_{SMGWa}(c0_2)$ now applies, which makes the dependence relation $dep_{SMGWa}(value0_c, \{c0_1, c0_2\})$ possible. Hence, the property $HasAcc_{SMGWa}(value0_c)$ holds and the negation consequently does not. Consequently, the privacy goal is not achieved in this model if one considers colluding attackers.

6.2 Procedure 2

Figure 4 shows the architecture of Procedure 2 (cf. section 5.2). The same simplifications as before apply. As described above, in this version a Key Management Authority distributes the keys. Due to the introduced encryption we now have more variables. The variables $k0_a, k0_b, k0_c$ denote the keys for the smart meters and $k0$ the corresponding key of the service provider of the next level.

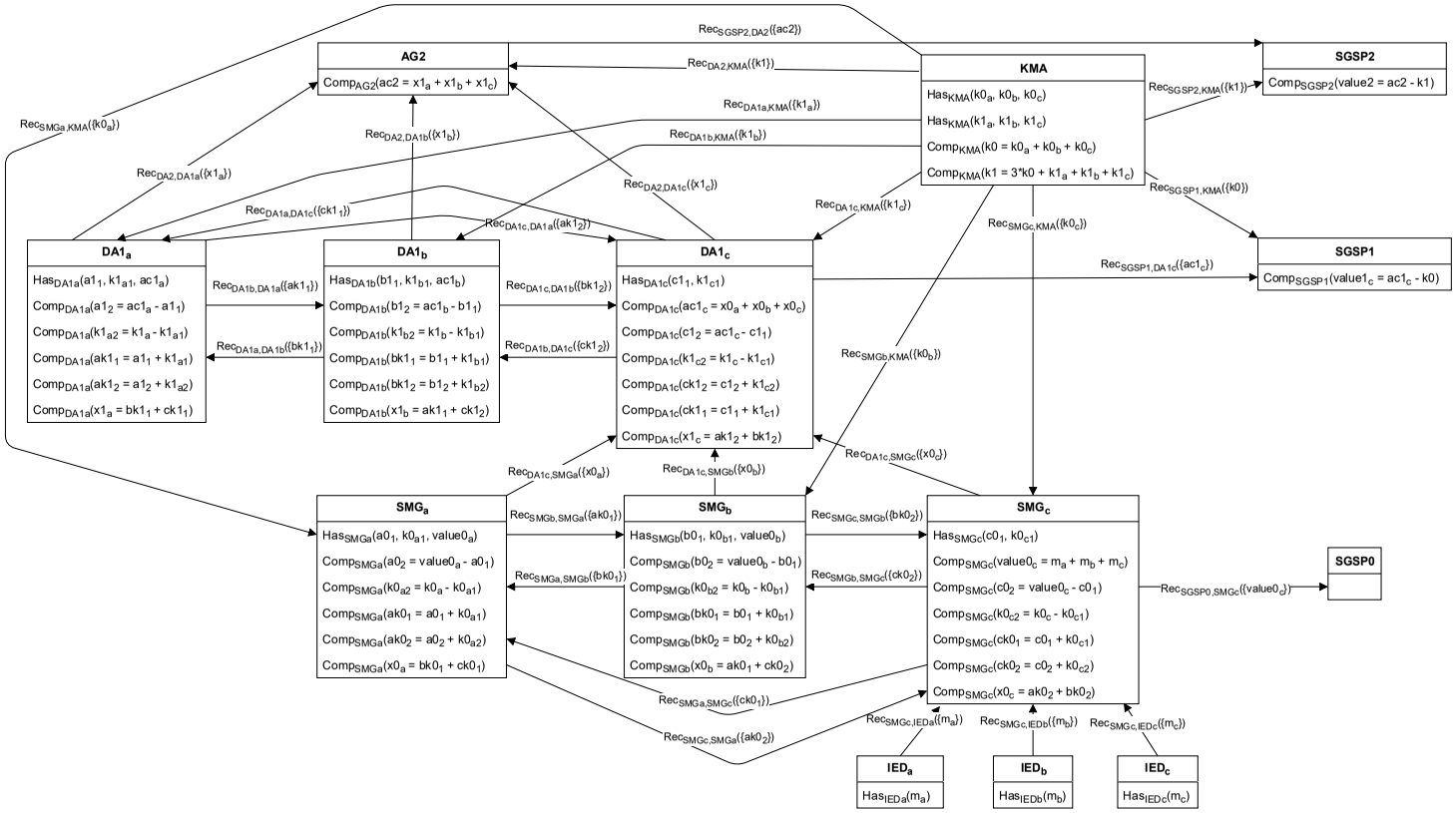

Fig. 4. Graphical representation of the Procedure 2 in the formal description language of CAPVerDE.

Accordingly, $k1_a, k1_b, k1_c$ are the keys for the street level aggregators and $k1$ is given to the service provider on the district level. In addition, the variables $ac1_a, ac1_b, ac1_c, ac2$ are introduced for intermediate results.

In this procedure our privacy goal is to guarantee that the aggregators cannot read the aggregated values which are intended for the respective service providers.

Non-Colluding Attackers The property $\phi_{2n} = \neg HasAcc_{AG1c}(value1_c)$ can be verified and produces a long trace. Essentially, the component $AG1c$ has no explicit data flow relations to obtain $value1_c$ and thus can only access it based on dependence relations. The only possible one is $dep_{AG1c}(value1_c, \{ac1_c, k0\})$. Rule **H3** applies for the aggregated encrypted sum $ac1_c$. For the service provider key $k0$, however, the component $AG1c$ can only use implicit data flow. Only two dependence relations exist, of which the first is a circle as $dep_{AG1c}(k0, \{ac1_c, value1_c\})$ is derived from the same equation as the previously used one. The second one is $dep_{AG1c}(k0, \{k0_c, k0_b, k0_a\})$, which needs the three keys from the SMGWs. $k0_c$ cannot be obtained via explicit data flow and thus $AG1c$ has to use dependence relations, again. Apart from the circle, only $dep_{AG1c}(k0_c, \{k0_c1, k0_c2\})$ is applicable. The only non-circular dependence relation for $k0_c1$ is $dep_{AG1c}(k0_c1, \{c0_1, k0_c2\})$. $dep_{AG1c}(c0_1, \{value0_c, c0_1\})$ is the only applicable implicit relation. However, we have shown above that $HasAcc_{SMGWa}(value0_c)$ does not hold for the architecture of Procedure 1. The same holds for this architecture as no non-circular dependence relation exists that allows $AG1c$ to obtain $value0_c$). Thus, the verification of the property $HasAcc_{AG1c}(value1_c)$ fails and the negation can be successfully verified.

Colluding Attackers Here we consider an additional collusion relation of the form $Rec_{SMGWa,SMGWb}(\{ck0_2\})$. The property $\phi_{1c} = \neg HasAcc_{SMGWa}(value0_c)$ is again to be verified. We omit the verification steps that are identical to the ones of Procedure 1. We start at dependence relation $dep_{SMGWa}(value0_c, \{c0_1, c0_2\})$. $c0_1$ can only be obtained via $dep_{SMGWa}(c0_1, \{ck0_1, k0_c1\})$. While Rule **H2** applies for $ck0_1$, $k0_c1$ can only be derived via accessing $c0_1$, which we are currently trying to obtain, or $k0_c$. The latter cannot be obtained by $SMGW_a$ without having $k0$ (circle again). Therefore, the verification of $HasAcc_{SMGWa}(value0_c)$ fails and we successfully verify ϕ_{1c}.

The proposed architecture achieves data confidentiality in the presence of honest-but-curious data aggregators as well as non-colluding Nano-Grid neighbors. It is not possible to deduce individual supply or demand from the aggregated values.

7 Conclusion

The proposed scheme describes an architecture for data aggregation and dissemination in Smart Grids supported by distributed Fog nodes. The architecture is

topology-aware in such that a grid-based hierarchy is built. A comprehensive modeling approach has been selected and demonstrated, which aims to enable Smart Grid services to transparently explain and proof their confidentiality. On the infrastructural layer, the scheme describes machine-to-machine communication which assures privacy-preserving aggregation of smart meter readings and dissemination between the SMGWs and possible Smart Grid Service Providers. We model the proposed scheme and show that its integrity and privacy-properties can be validated by CAPVerDE, an automated prover.

7.1 Future Work

In this paper, we have shown that general privacy properties of data flows in the context of Smart Grids can be modeled and proven using automated solvers. As a next step, we elaborate a systematization of aspects that need to be protected from unauthorized access and curious actors. We want to especially look into the protection of meta data and how communication relationships between the different actors can be obfuscated. Examples for such meta data could be the origin of readings, the recipient Smart Grid Service Providers, but also the type of reading (topic) and frequency of transmission.

The proposed scheme is efficient regarding the computational effort, but can still be optimized with respect to the messaging overhead. Anonymity and privacy-respecting data flows always have to evaluate between the price in efficiency and the return in privacy. For future work, we will evaluate other hierarchical aggregation schemes to conduct a systematic network load comparison.

Regarding the key management, for privacy reasons it might be beneficial to investigate distributed schemes instead of using a centralized key management authority. This challenging research field is another follow-up activity in the journey towards the development of truly privacy-respecting Smart Grids.

References

1. Abeykoon, V., Kankanamdurage, N., Senevirathna, A., Ranaweera, P., Udawalpola, R.: Real time identification of electrical devices through power consumption pattern detection. Pervasive Comput **10**(1), 40–48 (2016)
2. Antignac, T., Le Métayer, D.: Privacy Architectures: Reasoning about Data Minimisation and Integrity. In: Int. Workshop on Sec. and Trust Management. pp. 17–32. Springer (2014)
3. Antignac, T., Le Métayer, D.: Trust Driven Strategies for Privacy by Design. In: IFIP Int. Conf. on Trust Management. pp. 60–75. Springer (2015)
4. Antoniadis, I.I., Chatzidimitriou, K.C., Symeonidis, A.L.: Security and Privacy for Smart Meters: A Data-Driven Mapping Study. In: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). pp. 1–5 (Sep 2019)
5. Asghar, M.R., Dán, G., Miorandi, D., Chlamtac, I.: Smart Meter Data Privacy: A Survey. IEEE Communications Surveys Tutorials **19**(4), 2820–2835 (Fourthquarter 2017)

6. Backes, M., Hritcu, C., Maffei, M.: Automated Verification of Remote Electronic Voting Protocols in the Applied pi-Calculus. In: Comput. Sec. Found. Symp. pp. 195–209. IEEE (2008)
7. Barth, A., Datta, A., Mitchell, J.C., Nissenbaum, H.: Privacy and Contextual Integrity: Framework and Applications. In: Symp. on Sec. and Priv. p. 184–198. IEEE (2006)
8. Barthe, G., Danezis, G., Grégoire, B., Kunz, C., Zanella-Béguelin, S.: Verified Computational Differential Privacy with Applications to Smart Metering. In: 2013 IEEE 26th Computer Security Foundations Symposium. pp. 287–301 (Jun 2013)
9. Bavendiek, K., Adams, R., Schupp, S.: Privacy-Preserving Architectures with Probabilistic Guaranties. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). pp. 1–10. IEEE (2018)
10. Bavendiek, K., Mueller, T., Wittner, F., Schwaneberg, T., Behrendt, C.A., Schulz, W., Federrath, H., Schupp, S.: Automatically Proving Purpose Limitation in Software Architectures. In: IFIP International Conference on ICT Systems Security and Privacy Protection. pp. 345–358. Springer (2019)
11. Bohli, J.M., Gruschka, N., Jensen, M., Iacono, L.L., Marnau, N.: Security and Privacy-Enhancing Multicloud Architectures. IEEE Transactions on Dependable and Secure Computing **10**(4), 212–224 (2013)
12. Danezis, G., Fournet, C., Kohlweiss, M., Zanella-Béguelin, S.: Smart meter aggregation via secret-sharing. In: Proceedings of the First ACM Workshop on Smart Energy Grid Security. pp. 75–80. SEGS '13, Association for Computing Machinery, Berlin, Germany (Nov 2013)
13. Delaune, S., Ryan, M., Smyth, B.: Automatic Verification of Privacy Properties in the Applied Pi Calculus. In: IFIP Int. Conf. on Trust Management. pp. 263–278. Springer (2008)
14. Derryberry, J.: Compiling an Honest but Curiuos Protocol (5 2003), https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-876j-advanced-topics-in-cryptography-spring-2003/lecture-notes/lec050703.pdf
15. Dimitriou, T., Awad, M.K.: Secure and scalable aggregation in the smart grid resilient against malicious entities. Ad Hoc Networks **50**, 58–67 (Nov 2016)
16. Diovu, R.C., Agee, J.T.: Enhancing the security of a cloud-based smart grid AMI network by leveraging on the features of quantum key distribution. Transactions on Emerging Telecommunications Technologies **30**(6), e3587 (2019)
17. Efthymiou, C., Kalogridis, G.: Smart Grid Privacy via Anonymization of Smart Metering Data. In: Int. Conf. on Smart Grid Commun. (SmartGridComm). pp. 238–243. IEEE (2010)
18. Eibl, G., Engel, D.: Differential privacy for real smart metering data. Computer Science - Research and Development **32**(1), 173–182 (Mar 2017)
19. Fan, C.I., Huang, S.Y., Lai, Y.L.: Privacy-Enhanced Data Aggregation Scheme Against Internal Attackers in Smart Grid. IEEE Transactions on Industrial Informatics **10**(1), 666–675 (Feb 2014)
20. Ferrag, M.A., Maglaras, L.A., Janicke, H., Jiang, J., Shu, L.: A systematic review of data protection and privacy preservation schemes for smart grid communications. Sustainable Cities and Society **38**, 806–835 (Apr 2018), http://www.sciencedirect.com/science/article/pii/S2210670717308399
21. Finster, S., Baumgart, I.: Privacy-Aware Smart Metering: A Survey. IEEE Communications Surveys Tutorials **17**(2), 1088–1101 (Secondquarter 2015)
22. Fournet, C., Kohlweiss, M., Danezis, G., Luo, Z., et al.: ZQL: A Compiler for Privacy-Preserving Data Processing. In: USENIX Sec. Symp. pp. 163–178 (2013)

23. Gaboardi, M., Haeberlen, A., Hsu, J., Narayan, A., Pierce, B.C.: Linear Dependent Types for Differential Privacy. In: ACM SIGPLAN Notices. vol. 48, pp. 357–370 (2013)
24. General Data Protection Regulation: Regulation (eu) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46 (GDPR). Official Journal of the European Union (OJ) **59**(1-88), 294 (2016)
25. Greveler, U., Justus, B., Loehr, D.: Multimedia content identification through smart meter power usage profiles. In: Computers, Privacy and Data Protection. p. 8. CPDP, Brussels, Belgium (Feb 2012)
26. Gürses, S., Troncoso, C., Diaz, C.: Engineering Privacy by Design. Computers, Privacy & Data Protection **14**(3), 25 (2011)
27. Halpern, J.Y., Van Der Meyden, R., Vardi, M.Y.: Complete Axiomatizations for Reasoning about Knowledge and Time. SIAM J. on Computing **33**(3), 674–703 (2004)
28. Hoepman, J.H., Hubbers, E., Jacobs, B., Oostdijk, M., Schreur, R.W.: Crossing Borders: Security and Privacy Issues of the European e-Passport. In: International Workshop on Security. pp. 152–167. Springer (2006)
29. Jawurek, M., Johns, M., Kerschbaum, F.: Plug-In Privacy for Smart Metering Billing. In: Fischer-Hübner, S., Hopper, N. (eds.) Privacy Enhancing Technologies. pp. 192–210. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2011)
30. de Jonge, W., Jacobs, B.: Privacy-Friendly Electronic Traffic Pricing via Commits. In: International Workshop on Formal Aspects in Security and Trust. pp. 143–161. Springer (2008)
31. Kalogridis, G., Efthymiou, C., Denic, S.Z., Lewis, T.A., Cepeda, R.: Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures. In: 2010 First IEEE International Conference on Smart Grid Communications. pp. 232–237 (Oct 2010)
32. Kart, F., Miao, G., Moser, L.E., Melliar-Smith, P.: A Distributed e-Healthcare System Based on the Service Oriented Architecture. In: Int. Conf. on Services Computing. pp. 652–659. IEEE (2007)
33. Kim, J., Le, T.T.H., Kim, H.: Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature. Computational Intelligence and Neuroscience **2017**, 22 (Oct 2017), https://www.hindawi.com/journals/cin/2017/4216281/
34. Kremer, S., Ryan, M.: Analysis of an electronic voting protocol in the applied pi calculus. In: European Symp. on Programming. pp. 186–200. Springer (2005)
35. Kursawe, K., Danezis, G., Kohlweiss, M.: Privacy-Friendly Aggregation for the Smart-Grid. In: Fischer-Hübner, S., Hopper, N. (eds.) Privacy Enhancing Technologies. pp. 175–191. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22263-4_10
36. Li, M., Lou, W., Ren, K.: Data security and privacy in wireless body area networks. IEEE Wireless Commun. **17**(1), 51–58 (2010)
37. Lodge, T., Crabtree, A., Brown, A.: Developing GDPR Compliant Apps for the Edge. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 313–328. Springer (2018)
38. Maffei, M., Pecina, K., Reinert, M.: Security and privacy by declarative design. In: Comput. Sec. Found. Symp. pp. 81–96. IEEE (2013)

39. Martin-Martínez, F., Sánchez-Miralles, A., Rivier, M.: A literature review of Microgrids: A functional layer based classification. Renewable and sustainable energy reviews **62**, 1133–1153 (2016)
40. Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., Irwin, D.: Private memoirs of a smart meter. In: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building. pp. 61–66. BuildSys '10, Association for Computing Machinery, Zurich, Switzerland (Nov 2010). https://doi.org/10.1145/1878431.1878446, https://doi.org/10.1145/1878431.1878446
41. Mrabet, Z.E., Kaabouch, N., Ghazi, H.E., Ghazi, H.E.: Cyber-security in smart grid: Survey and challenges. Computers & Electrical Engineering **67**, 469–482 (Apr 2018). https://doi.org/10.1016/j.compeleceng.2018.01.015, http://www.sciencedirect.com/science/article/pii/S0045790617313423
42. R. C. Green, L.W., Alam, M.: High performance computing for electric power systems: Applications and trends. In: 2011 IEEE Power and Energy Society General Meeting. pp. 1–8. IEEE, Detroit, Michigan, USA (July 2011)
43. Rehmani, M.H., Davy, A., Jennings, B., Assi, C.: Software Defined Networks-Based Smart Grid Communication: A Comprehensive Survey. IEEE Communications Surveys Tutorials **21**(3), 2637–2670 (thirdquarter 2019). https://doi.org/10.1109/COMST.2019.2908266
44. Simmhan, Y., Aman, S., Kumbhare, A., Liu, R., Stevens, S., Zhou, Q., Prasanna, V.: Cloud-Based Software Platform for Big Data Analytics in Smart Grids. Computing in Science Engineering **15**(4), 38–47 (July 2013)
45. Stübs, M., Ipach, H., Becker, C.: Topology-aware distributed smart grid control using a clustering-based utility maximization approach. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing. pp. 1806–1815 (2020)
46. Stübs, M., Posdorfer, W., Momeni, S.: Blockchain-based multi-tier double-auctions for smart energy distribution grids. In: 2020 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE (2020)
47. Sultan, S.: Privacy-preserving metering in smart grid for billing, operational metering, and incentive-based schemes: A survey. Computers & Security **84**, 148–165 (Jul 2019). https://doi.org/10.1016/j.cose.2019.03.014, http://www.sciencedirect.com/science/article/pii/S0167404818303675
48. Van Aubel, P., Poll, E.: Smart metering in the Netherlands: what, how, and why. International Journal of Electrical Power & Energy Systems **109**, 719–725 (2019)
49. Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., Vasilakos, A.V.: Security and privacy for storage and computation in cloud computing. Inform. Sciences **258**, 371–386 (2014)

A Verification Trace

Figure 5 shows an excerpt of the verification trace of the property ϕ_{1n}. First a negation rule is applied and then the rules **H1** to **H4** are applied in order. Lines 5–10 show that no explicit data flow was found, i.e., component $SMGWa$ does not have the variable $value0_c$. Starting from line 11, CAPVerDE checks whether the component has the ability to compute the variable implicitly. Two matching dependence relations exist that allow deriving $value0_c$: $dep_{SMGWa}(value0_c, \{m_a, m_b, m_c\})$ and $dep_{SMGWa}(value0_c, \{c0_1, c0_2\})$. The first

```
1  Current property to prove: NOT HasAcc_SMG_a(value0_c)
2  Rule I_neg applied for statement:
3  Therefore trying to verify new statement:
4      Current property to prove: HasAcc_SMG_a(value0_c)
5      Trying Rule H1...
6      Rule H1 not applicable
7      Trying Rule H2...
8      Rule H2 not applicable
9      Trying Rule H3...
10     Rule H3 not applicable
11     Trying Rule H4...
12     Found Dep_SMG_a(value0_c,{m_a,m_b,m_c})
13         Current property to prove: HasAcc_SMG_a(m_a)
14         Trying Rule H1...
15         Rule H1 not applicable
16         Trying Rule H2...
17         Rule H2 not applicable
18         Trying Rule H3...
19         Rule H3 not applicable
20         Trying Rule H4...
21         Found Dep_SMG_a(m_a,{value0_c,m_b,m_c})
22             Current property to prove: HasAcc_SMG_a(value0_c)
23             Loop detected
24         Rule H4 not applicable
25         HasAcc_SMG_a(m_a) does not hold
26     Found Dep_SMG_a(value0_c,{c0_1,c0_2})
27         Current property to prove: HasAcc_SMG_a(c0_1)
28         Trying Rule H1...
29         Rule H1 not applicable
30         Trying Rule H2...
31         Rule H2 applied for statement: HasAcc_SMG_a(c0_1)
32         HasAcc_SMG_a(c0_1) holds
33         Current property to prove: HasAcc_SMG_a(c0_2)
34         Trying Rule H1...
35         Rule H1 not applicable
36         Trying Rule H2...
37         Rule H2 not applicable
38         Trying Rule H3...
39         Rule H3 not applicable
40         Trying Rule H4...
41         Found Dep_SMG_a(c0_2,{value0_c,c0_1})
42             Current property to prove: HasAcc_SMG_a(value0_c)
43             Loop detected
44         Rule H4 not applicable
45         HasAcc_SMG_a(c0_2) does not hold
46     Rule H4 not applicable
47     HasAcc_SMG_a(value0_c) does not hold
48 Rule I_neg applied for statement: NOT HasAcc_SMG_a(value0_c)
49
50 Property verification successful!
```

Fig. 5. Verification trace explaining the rule application of CAPVerDE.

is used in lines 12–25 but $SMGWa$ can neither explicitly nor implicitly get access to the variable m_a. Lines 26–45 show the trace of the second dependence relation. The component does have access to the variable $c0_1$ but cannot obtain the also necessary $c0_2$. As there are no more means to obtain variable $value0_c$ left, the property $HasAcc_{SMGWa}(value0_c)$ does not hold, hence the negation does (line 48).

The verification of the property $\phi_{2n} = \neg HasAcc_{AG1c}(value1_c)$, however, fails. Due to the relation $Comp_{AG1c}(value1_c = x0_a + x0_b + x0_c)$ the trace is very short. Rule **H3** applies and consequently $HasAcc_{AG1c}(value1_c)$ holds. Therefore, the negation does not hold and our data protection goal of protecting the aggregated measures against the next level aggregators is not satisfied in this architecture.