

Plug-in privacy for Smart Metering billing

Marek Jawurek, Martin Johns, Florian Kerschbaum
firstname.lastname@sap.com

SAP Research

Abstract. Traditional electricity meters are replaced by Smart Meters in customers' households. Smart Meters collect fine-grained utility consumption profiles from customers, which in turn enables the introduction of dynamic, time-of-use tariffs. However, the fine-grained usage data that is compiled in this process also allows to infer the inhabitant's personal schedules and habits. We propose a privacy-preserving protocol that enables billing with time-of-use tariffs without disclosing the actual consumption profile to the supplier. Our approach relies on a zero-knowledge proof based on Pedersen Commitments performed by a plug-in privacy component that is put into the communication link between Smart Meter and supplier's back-end system. We require no changes to the Smart Meter hardware and only small changes to the software of Smart Meter and back-end system. In this paper we describe the functional and privacy requirements, the specification and security proof of our solution and give a performance evaluation of a prototypical implementation.

1 Introduction

1.1 Motivation

Smart Metering has been mandated by EU directive 2009/72/EC and promoted by the US Energy Independence and Security Act of 2007 and Smart Meter roll-outs have begun all over the world [1]. Smart Meters record a fine-grained consumption profile of a certain service (electricity, heat or water) and report it to the supplier of the service who bills the customer accordingly. Traditionally only a single, compiled value for a whole reporting period has been reported to the supplier (e.g., the total consumed electrical energy of the last year). In contrast, Smart Meters transmit a detailed set of many data points which document consumption for short time intervals (e.g. every 15 minutes). This enables the suppliers to introduce more dynamic pricing schemes and to collect precise data about their customer base's usage patterns.

Besides the technical motivation, also legal reasons come into play in respect to the current push towards Smart Metering: For instance, in Germany starting October 2010 suppliers must offer either time or load dependent tariffs (see §40 [20]). These tariffs necessarily require Smart Meters with fine-grained consumption recording.

However, such detailed data has privacy implications: A listening third party, the supplier or even an employee of the supplier could learn the consumption

behavior of a customer and might use this information maliciously for other purposes than intended (e.g, to learn behavioral patterns, such as sleep/wake cycles or vacation time, of a given customer based on his energy usage). Recently, customers have become aware of the potential privacy implications of such consumption profiles. In the Netherlands Smart Meter roll-outs have been stopped because of the public outcry about the invasion of customer privacy [15].

Grid operators and suppliers now face a dilemma: On the one hand, they need to implement Smart Metering for legal and technical reasons. But, on the other hand, they face on-going problems in respect to public acceptance of the technology due to the outlined privacy problems.

1.2 Our solution

We provide a solution to this conflict by introducing a new consumption profile reporting protocol for time-of-use tariffs. We introduce a plug-in privacy component into the standardized Smart Meter / Meter Data Management (MDM) reporting communication link. This component hides the actual consumption profile from the MDM and therefore also from the supplier. We require only small changes compared to current Smart Meter reporting. The plug-in privacy component intercepts Smart Meter readings, then uses tariff information provided externally (over the Internet or by the MDM) to calculate the billing amount and sends only the resulting billing amount to the MDM. A Zero-Knowledge Proof ensures the correctness of the calculation.

The advantages of our approach are the following:

1. The Smart Meter's hardware complexity remains the same, because all calculations are conducted by the stand-alone plug-in component. Such a plug-in component can be realized by off-the-shelf computing hardware like a router or Wifi access point or even by software running on a standard personal computer.
2. The supplier does not have to trust the plug-in privacy component. The privacy component's output suffices to check whether it calculated the final billing amount honestly and correctly, i.e. based on the correct tariff and on the correct readings provided by the Smart Meter. Therefore the privacy component does not require hardware-protected components and can be quite simple and cheap. The correct operation of the privacy component can be verified only by its output.
3. Plaintext, fine-grained consumption profiles never even leave the household, if a privacy component is used. This prevents any abuse of this data, either by intercepting it in transit, by leakage in the MDM systems or by the MDM's operator himself. It also spares the MDM expensive security measures for the protection of the massive amount of privacy related data – the consumption profiles of his customers.

1.3 Paper outline

The remainder of this paper is structured as follows: Section 2 motivates the problem, gives a short introduction into Smart Metering and its privacy problems and defines our problem statement. In Section 3 we describe the underlying cryptographic method of our solution before we explain the setup of our solution, the specification of the protocol and its security analysis in Section 4. We evaluate a prototypical implementation in Section 5. Furthermore we show how our protocol might fit into existing Smart Meter communication protocols and how it fulfills the stakeholder requirements. Finally, we give an overview of related work in Section 6, provide an outlook on future work in Section 7 and conclude with a summary in Section 8.

2 Smart Metering’s implications for Privacy

2.1 Naming conventions

Before we explore the Smart Meter billing process and deduct its privacy implications, we briefly specify the terms used in the rest of this paper:

Customer: The term ”customer” represents the household, family or person that receives the service from a supplier.

Supplier: The term ”supplier” stands as placeholder for all companies that cooperate in order to provide the service to customers and also want to subsequently invoice the customers for this service.

Consumption profile: The term ”consumption profile” stands for the consumption data collected by Smart Meters for service in a certain interval over a certain period of time. This is applicable to many utilities (electricity, water, heat, gas, etc.).

Back-end system: Usually, the Smart Meter is directly connected to a MDM which just collects consumption data. Tariffs, are then applied in the supplier’s billing systems where the data is subsequently transported to. In this paper, ”back-end system” (BS) stands for the collection of all IT-systems that collect consumption profiles and use them to calculate the invoice for the customer based on tariffs.

Tariff: The term ”tariff” stands for the price schema, i.e., the price of service consumption at a specific interval. In the following we restrict ourselves to a time-of-use pricing scheme, but our protocol could also handle load-dependent billing with little modification.

2.2 Smart metering billing

Smart Metering refers to the collection of consumption profiles at customer’s households with the help of so called Smart Meters (SM). Smart Meters measure electricity consumption in households and communicate their readings at regular intervals to the back-end system. Alternatively, the back-end system can also

query the Smart Meter for its data (pull). A Trusted Platform Module (TPM) in the Smart Meter holds key material and creates signatures over the data to ensure authenticity and integrity until it arrives at the back-end system. There the consumption profile and the tariff data from the respective customer’s contract are used to calculate the price the customer has to pay for the time period covered by the profile. Figure 1 displays the usual Smart Meter setup.

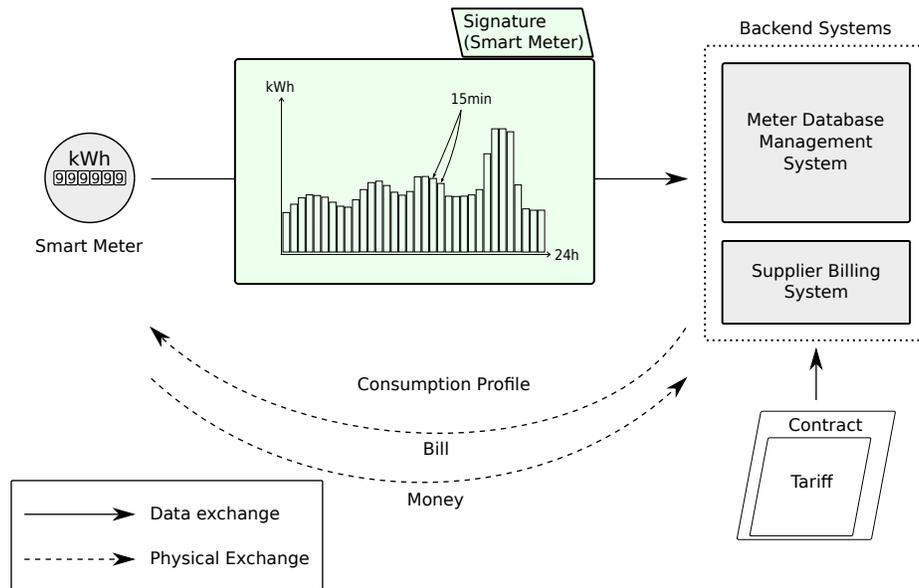


Fig. 1. Traditional setup of Smart Meter and back-end system

2.3 Privacy concerns

Smart Metering has encountered massive privacy concerns from media [16], data privacy experts [8] and consumers [15]. The fact that whole consumption profiles of households are transmitted to and stored by suppliers is troubling w.r.t. customer privacy. Data confidentiality can be easily protected in transit between SM and BS. However, their storage at the suppliers’ IT-systems still endangers customer privacy. Depending on resolution and the availability of different services’ profiles (e.g. water, heat, electricity) one can read the profile and ”see” more or less clearly what happens in the household: For instance, when family members wake up (light switched on), whether they shower in the morning (water, heat, and electricity for water heater), whether they drink hot beverages

with their breakfast and when or if they leave for work or school. Furthermore, the frequency of washing and drying clothes, cooking or the amount of time the TV is turned on can be inferred. For further research on what electricity consumption profiles tell about the inhabitants see [4], [13], [14], [24] or [29].

These inferences make consumption profiles very privacy-sensitive data and these profiles might even have value in the advertising market, for instance. On one hand, disgruntled employees or external attackers might attempt to steal it for profit or out of malice. On the other hand, the supplier could seek subsidiary revenues by selling this data himself. Depending on the local jurisdiction, this might even be legal.

The important point is, that currently there are no reliable, technical measures in place to prevent abuse of consumption profiles. Merely organizational measures, policies or laws sanction the abuse of privacy related data but require a trace or proof of abuse and do not prevent it in the first place.

2.4 Problem statement

The problem we tackle in this paper is to enable suppliers to do billing using Smart Metering data without actually receiving privacy related data.

Supplier’s requirements The supplier’s requirement regarding consumption profiles is the ability to reliably use the data in the consumption profile to calculate the customer’s bill for received electricity. The consumption profile V is a vector of n values v_i that represent the amount of utility used in the interval i of one day. The time-of-use tariff T is a vector of n t_i where interval i is priced with t_i . t_i and v_i are integers. Then the formula for calculating the time-of-use price for consumption of one day is

$$P(V, T) = \sum_{i=0}^n t_i * v_i$$

It is crucial for the supplier that the consumption profiles are accurate and trustworthy. Clearly, a customer might be inclined to report lower consumption than actually consumed, because it lowers his bill. Therefore the Smart Meters are equipped with the TPM in order to ensure that the reported consumption profiles are trustworthy and reliable.

Customers’ requirements In addition to the requirements of traditional metering (accuracy of the bill), a customer of Smart Metering is concerned about his privacy. The less information is leaked by the customer, the better for him. We strive for ideal privacy, i.e. the view of the consumption profile by the supplier is indistinguishable from a uniformly chosen consumption profile with the same price, i.e. supplier obtains no additional information to the price. Furthermore, complicated tariffs will necessitate a way for customers to verify their bill in a trustworthy manner. Being able to do so, without relying on the suppliers

billing systems, is a secondary requirements that customers will have with Smart Metering.

Infrastructure constraints A major constraint for the infrastructure investments in Smart Metering is cost. Suppliers have to replace conventional meters in every household with a new Smart Meter. This is a significant amount of money for a complete roll-out even for a utilities' provider. Therefore every technology built into a Smart Meter faces scrutiny w.r.t. to costs.

This also includes the security measures like TPMs and secure storage. The development and verification of a secure TPM is very expensive and therefore it is common practice to keep its functionality minimal. One naive approach to privacy-preserving billing of consumption profiles would be to calculate the price in the TPM itself. But this would require that tariff information is retrieved and verified by the TPM. In turn, this would require adding an input communication channel and module to the TPM and would consequently increase the costs for building and verifying the TPM considerably.

Legal constraints Depending on the jurisdiction, metering can be subject to legal requirements. In Germany, for instance, metrology laws [21] govern require a certain degree accuracy of the meter and measurements and the tamperproofness of the meter. Privacy laws [22] require the confidentiality of readings to protect consumers' privacy. We translate this into the technical requirements of Smart Meter integrity and integrity and confidentiality of consumption profiles on the wire and in computer systems.

3 Pedersen Commitments

The core of our proposed solution (which we present in Sec. 4) relies on Pedersen Commitments [26]. In this section we briefly introduce the basics of this cryptographic method. For further information on the scheme please refer to [26].

A commitment is a cryptographic tool with two functions:

- $Commit(x, r) \rightarrow c$ takes as input a value x and a random number r . As output it produces the commitment c .
- $Open(c, x, r) \rightarrow \top/\perp$ takes as input a commitment c , a value x and a random number r . It outputs \top , if c is indeed a commitment to x and \perp , if not.

Commitments have two security properties:

- *Secret*: Given c it is hard to compute x .
- *Binding*: Given c , x and r it is hard to compute an $x' \neq x$ and r' , such that $Open(c, x', r') = \top$, i.e. c is a commitment for x' as well.

They are used in the following way: Alice chooses a value x . She computes a commitment c and sends it to Bob. Now, Alice and Bob may, for example, engage in some computation that depends on Alice’s input x , but where Alice may no longer change her mind. Alice opens her commitment and shows that everything was indeed computed according to the value x she choose at the beginning.

A typical example is fair coin flip. Alice chooses a random s and sends the commitment c of s to Bob. Bob chooses a random number t and sends it to Alice. Alice now opens her commitment. The fair coin flip is $x = s \oplus t$ (where \oplus denotes “exclusive-or”). If the commitment was not secret, Bob could choose x . If the commitment was not binding, Alice could choose x .

Pedersen commitments operate over a group \mathbb{G} . This group \mathbb{G} can be the same elliptic curves as used of EC-DSA in the secure hardware of the Smart Meter, but in our implementation we use the group \mathbb{Z}_p^* . Let g and h be two generators of \mathbb{G} . Pedersen commitments are computed as follows:

– *Commit*(x, r):

$$c = g^x h^r$$

– *Open*(c, x, r):

$$c \stackrel{?}{=} g^x h^r \hat{=} c \stackrel{?}{=} \text{Commit}(x, r)$$

The proofs of their security properties can be found in [26].

Pedersen commitments have another very useful property we exploit in this paper. They are homomorphic, i.e. a multiplication of two commitments results in a commitment to the sum of their committed values.

$$\text{Commit}(x, r)\text{Commit}(y, s) = \text{Commit}(x + y, r + s)$$

A commitment can also be multiplied by a plain factor y

$$\text{Commit}(x, r)^y = \text{Commit}(xy, ry)$$

Note that both operations change the commitment c , such that the binding security property is not violated. Instead one needs to open with the new input values of the commitment.

4 The private billing protocol

In this section we describe our privacy-preserving Smart Meter billing protocol. First we give a very abstract description in Section 4.1, then we provide the full specification in Section 4.2 and provide a security analysis in Section 4.3.

4.1 Components and specification

The main idea of our approach is that the plaintext consumption profiles never leave the household, but only after they have been processed by a pseudo-random one-way function. Therefore, ideal privacy is preserved. We propose to introduce

a privacy component (PC) into the communication link of the Smart Meter and the supplier’s back-end system. Its objective is to intercept reports of consumption profiles and to let only processed information pass-through. The PC is invisible to the SM and only the supplier will notice it: The PC directly interacts with the supplier’s systems and consumption profiles will look different if a PC is used. This setup is illustrated in Figure 2.

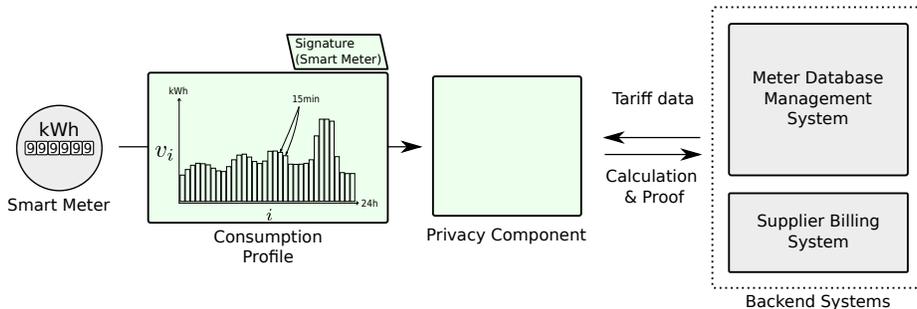


Fig. 2. Setup of proposed solution with intermediate privacy component

The major difference to a standard Smart Metering setup is that the price function $P(V, T)$ is not calculated at the supplier’s system. It is calculated in the PC which is supposed to be located in the household. For this, the PC intercepts the consumption profile and signed commitments sent to it by the Smart Meter and removes the plaintext consumption profile. Then the PC obtains the tariff information from the supplier and calculates the bill with the original consumption profile. It then presents the invoice, the signed commitments and a Zero Knowledge Proof to the supplier who verifies the bill’s validity using the homomorphic property of the used commitment scheme: The supplier determines the correctness of the bill by appropriate operations on the received signed commitments and the tariff. If the commitments can be verified on the presented bill, then the presented bill is trustworthy and correct. The homomorphic commitment scheme we use on the Smart Meter side is Pedersen Commitment [26] and is shortly outlined in Section 3.

4.2 Protocol specification

Initiation Initially, the SM and BS need to employ some signature scheme which allows the SM to secure the integrity of data sent to the BS. This is usually already the case with Smart Meters. They are either part of a PKI or both, the SM and the BS, have access to a symmetric key for a symmetric signing scheme. We denote such signing key with $\text{Sign}_{\text{priv}}$.

Secondly, the TPM in the Smart Meter must be able to use the Pedersen Commitment scheme (see Section 3) with public generators g and h . How keys

(or the public parameters, such as the generators) are distributed to Smart Meters is beyond the scope of this paper, but it is already common practice in on-going Smart Meter deployments.

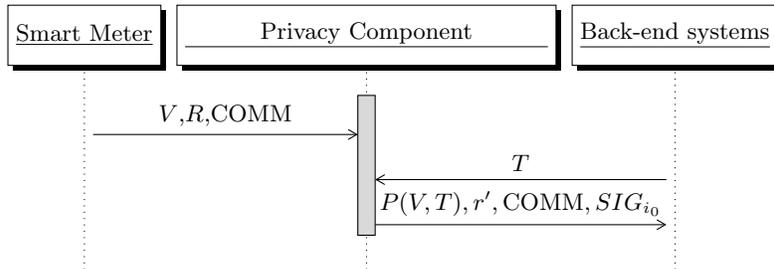


Fig. 3. Communication sequence

Consumption profile reporting and invoice calculation Figure 3 illustrates the communication that takes place between the different actors and the following enumeration of steps describes the protocol in detail:

1. The SM prepares a consumption profile to be reported to BS. The profile basically consists of a vector of consumption values $V = \{v_{i_0}, v_{i_1}, \dots, v_{i_n}\}$. v_{i_k} represents the energy consumption of the household in the interval i_k . i_k stands for the interval number, incremented since a fictive first interval, analogous to the definition of the UNIX time stamp.
2. For values in V the SM now creates commitments. The commitment of v_{i_k} is $Comm_{i_k} = Commit(v_{i_k}, r_{i_k})$. Where $Commit(a, r)$ stands for the Pedersen Commitment of a and a random value r with the generators g and h known to the TPM and the BS.
3. Now, the SM would like to send the data to the BS. Before that can happen, it protects the data from being manipulated on the way. It creates a signature SIG_{i_0} (with its signing key $Sign_{priv}$) over $(i_0, COMM)$ and sends it together with the vector V , the vector $COMM = \{Comm_{i_0}, Comm_{i_1}, \dots, Comm_{i_n}\}$ and the vector $R = \{r_{i_0}, r_{i_1}, \dots, r_{i_n}\}$ towards the supplier's back-end system.
4. The PC intercepts all the traffic between the SM and the BS.
5. The PC obtains the tariff vector $T = \{t_{i_0}, t_{i_1}, \dots, t_{i_n}\}$ from BS and performs the following two calculations:
 - (a) $P(V, T) = \sum_{k=i_0}^{i_n} v_k * t_k$ This is the actual price the customer has to pay for the reporting period represented by V .
 - (b) In addition it also calculates r' from the vector R it intercepted in step 4: $r' = \sum_{k=i_0}^{i_n} r_k * t_k$
6. The PC now sends $P(V, T), r', COMM, SIG_{i_0}$ to BS and has finished its work.

Verification These are the steps performed by the BS subsequently to the reporting in order to verify that the $P(V, T)$ was correctly calculated:

7. First of all, the BS verifies that the signature SIG_{i_0} over i_0 and the commitments is intact which means that the commitments it received has been signed by the TPM and stands for the next vector $V = \{v_{i_0}, v_{i_1}, \dots, v_{i_n}\}$ starting from i_0 .
8. BS now computes $COMM_{\text{Tariff}}$ with the $Comm_{i_0}$ it received in step 6 and the tariff vector T that it made available to PC in step 5:

$$COMM_{\text{Tariff}} = \prod_{k=i_0}^{i_n} Comm_k^{t_k}$$

9. Whether the $P(V, T)$ sent by the PC has been calculated truthfully with the correct v_i and t_i can now be verified by opening the aggregated commitment $COMM_{\text{Tariff}}$. For that, the BS uses $P(V, T)$ and the aggregate random value r' that it received in step 6.

$$\begin{aligned} & Open(COMM_{\text{Tariff}}, P(V, T), r') \\ &= COMM_{\text{Tariff}} \stackrel{?}{=} Commit(P(V, T), r') \end{aligned}$$

4.3 Analysis

Theorem 1 *Our private billing protocol is complete, sound and honest-verifier zero-knowledge.*

Proof. For completeness, i.e. if the PC truthfully computes the tariff, then the BS accepts, we observe the following equation:

$$\begin{aligned} & Commit(P(V, T), r') \\ &= Commit\left(\sum_{k=i_0}^{i_n} t_k v_k, \sum_{k=i_0}^{i_n} t_k r_k\right) \\ &= \prod_{k=i_0}^{i_n} Commit(t_k v_k, t_k r_k) \\ &= \prod_{k=i_0}^{i_n} Commit(v_k, r_k)^{t_k} \\ &= \prod_{k=i_0}^{i_n} Comm_k^{t_k} \\ &= COMM_{\text{Tariff}} \end{aligned}$$

It follows that $COMM_{\text{Tariff}}$ is a Pedersen commitment for $P(V, T)$ with the random number r' .

For soundness we prove that if the PC does not truthfully compute the tariff, then the BS must reject. That is given v_i , the PC cannot forge a view $Comm_i$, $P(V, T)$ and r' of the protocol that is accepted by the BS.

We will prove by contradiction. First, observe that we assume that the PC cannot forge the $Comm_i$ commitments, since they are signed by the TPM. Second, as follows from completeness, the subsequently computed $COMM_{\text{Tariff}}$ is a Pedersen commitment to $P(V, T)$ and r' . If, the PC could present $P'(V, T) \neq P(V, T)$ and r'' , such that $COMM_{\text{Tariff}} = \text{Commit}(P'(V, T), r'')$ is opened correctly, then this would be a contradiction to the binding property of Pedersen commitments as established in [26].

For honest-verifier zero-knowledge, we present a simulator of the view of the BS given only its input and output. The values $Comm_i$ and r' from the view of the protocol are uniformly and independently distributed in \mathbb{Z}_p^* . The tariff $P(V, T)$ is public output of the protocol (and input to the verification operation).

The signature $Sig_{i_0}(Comm_1, \dots, Comm_n)$ of the TPM cannot be trivially simulated, since the BS only holds the public key. Nevertheless, since it is only a signature of randomly distributed values, we could simulate it by inverting the signature verification operation on a random signature. This rather strange simulation is an artifact of our unconventional setup of proving having the PC compute on input from another party – the TPM. In a strict sense, the signature is not part of the Zero Knowledge Proof, since it is computed by the TPM and not the PC.

5 Implementation and Evaluation

In this section, we give details on our prototypical implementation, show how our component can be integrated in real world Smart Meter deployments, and discuss how our solution fulfills the functional and security requirements which were identified in Section 2.4.

5.1 Implementation of the core algorithm

We implemented an exemplary system to identify load on the respective hardware systems during the execution of our protocol. For this purpose, we modeled the SM, the PC and the BS in Java as much as necessary to execute our protocol. In our implementation the SM creates a consumption profile from 96 fake readings and enriches the profile with commitments over the individual readings. It generates Pedersen Commitments in \mathbb{Z}_p^* where p has bit length 1024, with respective bit lengths of 32 for the readings and the tariff, 160 bits for r and the generators $g = 2$, $h = 3$. We use the BigInteger class and its methods for representing and handling commitments and randoms. However, we used our own implementation of modular exponentiation with precomputed powers of g/h which is faster than the built-in method of the BigInteger class (see time differences in Table 1).

For the evaluation of performance we chose a Java Benchmarking framework explained in [7]. It provides us with a method to calculate means and standard deviations in a statistically sound manner, dealing with different Java specialties like dynamic optimization, resource reclamation and caching.

Table 2 displays the means and standard deviations of the respective calculations in the protocol. Execution times were measured on a Intel(R) Core(TM) i5 CPU M540 at 2.53GHz on a OpenJDK Runtime Environment (IcedTea6 1.8.1) on a Ubuntu 10.10 system.

Method	Mean	Standard deviation
BigInteger.modPow	174.373 ms	3.400 ms
Our implementation	126.396 ms	6.175 ms

Table 1. Creation of commitments in SM (96 commitments) with built-in BigInteger.modPow vs own implementation

Computation	Component	Mean	Standard deviation
Creation of commitments (step 2)	SM	126.396 ms	6.175 ms
Aggregation of randoms (step 5b)	PC	24.359 us	183.239 us
Aggregation of commitments (step 8)	BS	7.443 ms	1.736 ms
Opening of aggregated commitment (step 9)	BS	1.442 ms	984.345 us

Table 2. Execution time means and standard deviations for the different computations in the protocol

From the numbers in Table 2 one can see that most time is spent in the SM and the BS. The most expensive calculation is performed by the Smart Meter and its hardware, respectively the TPM performing the actual calculations. It is usually several scales inferior to our test system but we believe that the SM is able perform its part of the protocol in a timely manner [5]. After all, irrespective of other constraints, it has one day before it needs to perform the next protocol run.

On the supplier’s system side one has to take into account that the supplier’s systems will need to participate in several thousand instances of this protocol per day, one for every associated household. If we assume that the supplier buffers the received verification data of concurrent protocol instances it can spread verification (where all of its time is spent) over the course of a whole day. Then, one such system (with our hardware) should be able to handle approx. one million protocol instances per day. This could be further increased by only verifying a random choice of reported prices. This shows, that a supplier should be able to implement our protocol for millions of users with negligible resources.

Our protocol necessitates higher data volumes to be transported between the SM, the PC and the BS than in a naive reporting protocol. In the most basic reporting protocol (with our assumptions that readings are 32bits wide) the SM has to sent 96 values (per day) to the BS. In addition to that, our protocol requires commitments (of 1024bits) and randoms (of 160bits) to be sent from the SM to the PC for every reading. The PC then forwards only 96 commitments, one aggregated random and one price. Table 3 displays the data volumes of our protocol. In total, our protocol requires 27284bytes to be transported over communication links, 26900bytes upstream (SM over PC to BS) and 384bytes downstream from BS to PC per reporting period (day). Although, our protocol poses a significant overhead with approx. factor 70 over the naive reporting protocol those volumes can still be handled even by slow GPRS connections. In this analysis we disregarded additional readings, meta-data and signatures that would have to be transported in the naive protocol as well as in our protocol.

Description	Source	Destination	Volume in bytes
Reporting of 96 readings (naive reporting without privacy)	SM	BS	$96 * 4 = 384$
Reporting of 96 readings, commitments and randoms	SM	PC	$96 * (128 + 20 + 4) = 14592$
Transport of tariff information for 96 time slots per day	BS	PC	$96 * 4 = 384$
Forwarding of 96 commitments and aggregated random	PC	BS	$96 * 128 + 20 = 12308$
Total amount of data volume in privacy-preserving reporting	*	*	$14592 + 384 + 12308 = 27284$

Table 3. Data volumes of our protocol vs a naive reporting protocol without privacy

5.2 Integration in real world scenario

The technology used by the Smart Meter to communicate to the Back-end system dictates how a Privacy Component could be introduced into this setup. Wireless communication links like ZigBee or Wifi would require that the SM is explicitly configured to report to the PC instead of the BS. Wireless communication links like GPRS would implicate higher costs, as the SM would need to communicate with the PC and the PC would need to establish an additional connection to the BS. Wired communication links, on the other hand, allow our scheme to be introduced more easily. The PC would then interrupt the physical link and act as physical gateway between the SM and the BS.

Regarding the integration of our approach into current Smart Meter reporting protocols, we have identified two relevant application layer protocol specifications (from [9]) for SM to BS reporting: The universal DLMS/COSEM standard suite (IEC 62056 / EN 13757-1) [2] and the simple Smart Metering Language (SML) [10] specification. Landis+Gyr, recently nominated [31] leader in Smart Metering, supports DLMS with its Landis+Gyr ZMD100 AP/AS residential meters and SML with its E750 Industrial and Commercial Smart Meters.

The Smart Meter Language (SML) describes an application and presentation layer and Smart Meters operate either in a push (SM initiates) or pull (SM reacts) scenario. All data is encoded in either an SML request or SML response message. Encryption of SML messages on the application or presentation layer is not part of the SML specification.

DLMS/COSEM is an application layer protocol. DLMS specifies how one can talk about energy metering objects. Energy Metering objects are described by the COSEM specification. The standard does not dictate specific transport protocols. The Smart Meter operates as server and communication follows the pull-strategy from the view of the BS system. Read and write access is realized by transmitting respective COSEM objects in APDUs (Application Protocol Data Units). The server's application context determines whether APDUs are encrypted.

How the privacy component can be embedded in environments employing SML or DLMS/COSEM depends on a multitude of factors: The actual protocols used on the network/transport layers, the used push/pull strategy as well whether encryption is used. Those factors determine whether the PC acts as transparent or visible proxy, how it intercepts messages and whether it needs key material to decipher messages.

For SML we see a simple straightforward solution how to implement the privacy component on the application layer: In SML actual consumption profiles are sent as tables with one row for each recorded value. The columns can describe one entry further with entries like time of recording, error conditions and so on. The whole table but also individual columns of the table can be signed which allows us to fit our protocol into SML messages easily: For every query (pull-scenario) of consumption values the SM answers with a table with the columns: $i, v_i, Comm_i, r_i$. The SM signs all columns independently but the PC intercepts the SML response, deletes the columns v_i and r_i from the table and inserts $P(V, T), r'$ and COMM into the message. Sig_{i_0} is represented by the i 's and $Comm_i$'s columns' intact signatures. This only requires, that the part of the Smart Meter responsible for creating signatures also create commitments. The BS system will notice that columns v_i and r_i are missing and will therefore switch into a mode where it communicates with a privacy component and performs the verification part of our protocol. If a privacy component was not employed the whole table is transferred intact and the BS system performs its normal operation and stores the plaintext values in its database.

For DMLS/COSEM the approach works analogously but with COSEM objects and properties instead of SML tables. However, in DLMS/COSEM encryp-

tion could make it impossible for the PC to understand the intercepted APDUs. In such a case, the Smart Meter either needs to be reconfigured not to use encryption or to use the public key of the PC instead of the supplier's public key. This will allow the PC to read and manipulate the APDU and possibly re-encrypt it for the supplier with the supplier's public key.

5.3 Fulfillment of stakeholder's requirements

In Section 2.4 we listed requirements of the different stakeholders for Smart Metering. We will show in this Section how our approach fulfills these requirements.

- In Section 2.4 we mentioned that the supplier's requirement is the trustworthiness of reported consumption values. Our protocol fulfills this by providing a trustworthy price instead of individual consumption values. We have given a soundness proof of our Zero Knowledge proof for the correct calculation of the price.
- In Section 2.4 we also stated that the customer wants privacy-aware billing. Our approach achieves a privacy-aware billing with ideal privacy as the consumption profile never leaves the household unprocessed, but only the price. We have proven the zero knowledge property of our Zero Knowledge Proof, i.e. the supplier will learn nothing, but the price. And while our approach was primarily developed for the fulfillment of the privacy-property it can also provide means to the customer to make billing reproducible. The PC can store intercepted consumption profiles and display them, together with the calculated prices, to the customer. This way, the customer can have real-time information about his energy consumption and current costs and also verify the invoice eventually sent out by the supplier.
- The infrastructure requirement listed in Section 2.4 is a low-cost Smart Metering solution. Our approach achieves this by only minimal changes to the software of Smart Meters and supplier's back-end system. The privacy component itself is simple and untrustworthy. It can therefore be implemented in inexpensive hardware or even in software.
- Finally, in Section 2.4 we mention several requirements regarding the tamperproofness, accuracy and confidentiality of the Smart Meter. As our approach does not interfere with the Smart Meter's normal operation accuracy and tamperproofness of the Smart Meter are not changed. We conform to any regulation we are aware of.

In addition, the supplier might benefit from the use of a privacy component as well, as less privacy-related data has to be stored in his systems for legal retention periods. The supplier needs to store all data that he receives from the PC for being able to reproduce invoice calculation for a certain retention time but that data is not privacy-related. The commitment values do not disclose useful information and the only privacy-related data item is the final price. This reduces the supplier's need for special security measures of his systems against internal or external attackers.

Based on the discussion above, it is safe to conclude that our approach fulfills all identified requirements (see Section 2.4) for a privacy-respecting billing of Smart Metering consumption profiles. Furthermore, as shown in Sections 5.1 and 5.2, an implementation of our algorithm is suitable to be deployed on a large scale and fits well with existing standards and infrastructures.

6 Related work

General references concerning security aspects of Smart Metering: Abstract predictions about security and privacy challenges potentially coming with the evolution of the current grid to the Smart Grid are described in [17] while [11],[25] and [8] give more information on the topics of security, privacy and trust in Smart Grids/Smart Metering and Advanced Metering Infrastructures (AMI).

In [8] Ontario's (Canada) Information and Privacy Commissioner provides an overview of what the Smart Grid is, how it will affect electricity consumers and how their privacy might be at risk by the Smart Grid and Smart Metering. Furthermore she promotes the idea of building privacy into the Smart Grid from the start.

In [23] the authors first perform an informal threat analysis of Smart Metering and provide a sketch for an attested Smart Meter architecture. Using virtualization, mandatory network access control and trusted computing techniques this architecture enables multiple applications to use the Smart Meter hardware and to work in a privacy-preserving and integer manner. The article names applications for billing the customer very closely to the data origin (in the household) and applications that provide the consumer with a consumer portal. They achieve privacy-preserving Smart Metering billing by remote attestation of the billing software in the TPM of the Smart Meter. For that, in contrast to our solution, they require quite powerful hardware capable of virtualization and they have a higher attack surface due to a more complicated Smart Meter design.

Privacy aspects of smart meter-based billing: In [12] a privacy-preserving detection algorithm for leakages in electricity distribution has been proposed. By aggregation across several Smart Meters the developed algorithm protects individual meter readings while allowing grid operators to detect illegitimate/unknown load. Their approach does not allow individual billing, yet this is the main application of our paper.

Furthermore, in [6] a model for measuring privacy in Smart Metering is developed and subsequently two different solutions to privacy are presented: A Trusted Third Party-based approach, where individual consumption profiles are aggregated at the third party and only sums are communicated to the supplier. The other approach attempts to mask consumption profiles by adding randomness to the actual profile with an expectation of the random distribution of zero. In contrast to our solution, both of their approaches cannot handle billing

of time-of-use tariffs but only provide either sums or not-accurate profiles. Furthermore, our approach does not require a trusted third party and provides exact results for every computation (as required by some legislations).

Finally, in [27] also a twofold approach is presented: The first solution employs a sophisticated Trusted Platform Module (TPM) in the Smart Meter to obtain signed tariff data from the supplier and calculate a trustworthy bill. The second solution makes use of the electrical grid infrastructure as a third party to anonymize up-to-date consumption values sent out constantly by Smart Meters. Our approach can be distinguished as it only addresses billing but only requires a very simple TPM that creates commitments.

Concurrent work [28] also covers the aspect of billing in Smart Metering. It focuses on realizing different tariff types where we provide practical information on how our scheme can be combined with existing Smart Metering reporting protocols.

Similar applications of homomorphic commitments: The articles [3] and [18] describe how privacy-preserving electronic toll pricing can be implemented using commitments to protect the time and location of cars. On-board-units continuously collect their location, time and the price of the consumed service (the driven route). They send commitments to the traffic authority. Eventually, using random spot checking, the traffic authority challenges on-board-units to open their commitments for places and times where their cars have been recorded by cameras. In contrast to our scheme, they use commitments to mostly protect auxiliary information used for computing the price, i.e. the time and place. Our work uses commitments to hide the private information from which the price is directly computed on the commitments, i.e. the energy volume per time slot. Consequently, we never need to open any input data commitment.

Pedersen commitments: Due to their homomorphic properties Pedersen commitments are an effective means to verify the correctness of statistics computation. In [30] it has been applied in the outsourced database setting. Statistics and dot product computation can be useful for in many application areas. An example from the database community again is privacy-preserving data mining [32]. An example from the business software community is collaborative benchmarking [19].

Our work is the first in providing a very high degree of privacy for customers by not disclosing consumption profiles in time-of-use smart meter billing scenarios.

7 Future work

Dynamic, time-dependent billing is only one application of fine-grained consumption data. In addition, a profile of a household's energy consumption can be utilized by the supplier to create predictions of this household's energy demand in the future. Our proposed solution does not cover this usage of consumption

profiles. Realizing a privacy friendly method for calculating such predictions is subject to future research. However, one must realize that privacy and the ability to create predictions potentially conflict with each other and this conflict should be investigated further also in the field of Smart Metering.

8 Conclusion

In this paper we have proposed a protocol for privacy-preserving reporting of consumption profiles in a Smart Metering scenario by the use of a plug-in component. We have identified and analyzed the requirements of different stakeholders. Based on this analysis, we devised a billing scheme which allows privacy-related consumption profiles to remain within the household while preserving provable correctness of the billable amounts. The privacy sensitive data therefore is not susceptible to interception in transit or leakage in the supplier's back-end system.

We have provided the specification for the utilized components, for the introduction into a traditional Smart Metering setup, and for the communication and calculation during the three protocol stages (initialization, reporting, verification). After proving the soundness, completeness and zero-knowledge property of the verification, we investigated the execution times of our prototypical implementation and showed that it is a viable solution for Smart Metering hardware. Finally, we discussed how our protocol could be executed using existing Smart Metering reporting specifications and showed that our approach fulfills the previously identified stakeholder's requirements.

Our protocol is one step towards the idea of building privacy into the Smart Grid [8]. By preserving customer privacy we mitigate trust issues that privacy experts, the media and the public have raised about the privacy implications of Smart Metering.

References

1. Smart metering projects map. <http://maps.google.com/maps/ms?ie=UTF8&oe=UTF8&msa=0&msid=115519311058367534348.0000011362ac6d7d21187>, 2010.
2. D. U. Association. DLMS/COSEM: Device language message specification/companion specification for energy metering. <http://www.dlms.com/organization/index.html>, 2010.
3. J. Balasch, A. Rial, C. Troncoso, C. Geuens, B. Preneel, and I. Verbauwhede. Pretp: Privacy-preserving electronic toll pricing. In *19TH USENIX SECURITY SYMPOSIUM*, pages 63–78. USENIX Association, 2010.
4. G. Bauer, K. Stockinger, and P. Lukowicz. Recognizing the use-mode of kitchen appliances from their current consumption. In *EuroSSC*, pages 163–176, 2009.
5. P. Bichsel, J. Camenisch, T. Groß, and V. Shoup. Anonymous credentials on a standard java card. In *ACM Conference on Computer and Communications Security*, pages 600–610, 2009.
6. J.-M. Bohli, O. Ugus, and C. Sorge. A privacy model for smart metering. In *Proceedings of the First IEEE International Workshop on Smart Grid Communications (in conjunction with IEEE ICC 2010)*, 2010.

7. B. Boyer. Companion material for robust java benchmarking article series, 2010.
8. A. Cavoukian, J. Polonetsky, and C. Wolf. Smart privacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, 3(2):275–294, August 2010.
9. K. D. Craemer and D. Geert. Analysis of state-of-the-art smart metering communication standards. In *Young Researcher Symposium edition 2010, Leuven, Belgium*, 2010.
10. D. M. W. EMSYCON GmbH. Smart meter language specification (SML). January 2008.
11. M. O. G. Lenzini and W. Teeuw. Trust, security, and privacy for the advanced metering infrastructure. Technical report, July 2009.
12. F. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In *Proceedings of the 6th International Workshop on Security and Trust Management*, 2010.
13. G. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, dec 1992.
14. G. W. Hart. Residential energy monitoring and computerized surveillance via utility power flows. *IEEE Technology and Society Magazine*, June 1989.
15. W. Heck. Smart energy meter will not be compulsory. NRC Handelsblad, April 2009. http://www.nrc.nl/international/article2207260.ece/Smart_energy_meter_will_not_be_compulsory.
16. A. Jamieson. Smart meters could be 'spy in the home'. Telegraph (UK), October 2009. <http://www.telegraph.co.uk/finance/newsbysector/energy/6292809/Smart-meters-could-be-spy-in-the-home.html>.
17. M. Jawurek and M. Johns. Security challenges of a changing energy landscape. In *ISSE 2010, Securing Electronic Business Processes*, pages 249–259. Vieweg+Teubner, 2010.
18. W. Jonge and B. Jacobs. Formal aspects in security and trust. chapter Privacy-Friendly Electronic Traffic Pricing via Commits, pages 143–161. Springer-Verlag, Berlin, Heidelberg, 2009.
19. F. Kerschbaum. Practical privacy-preserving benchmarking. In *Proceedings of the 23rd IFIP International Information Security Conference*, volume 278 of *IFIP International Federation for Information Processing*, pages 17–31, Boston, MA, USA, 2008. Springer.
20. G. Legislation. Energiewirtschaftsgesetz, 2005.
21. G. Legislation. Gesetz über das Meß- und Eichwesen (Eichgesetz), 2008.
22. G. Legislation. Bundesdatenschutzgesetz, 2009.
23. M. Lemay, G. Gross, C. A. Gunter, and S. Garg. Unified architecture for large-scale attested metering. In *Hawaii International Conference on System Sciences. Big Island*. ACM, 2007.
24. M. A. Lisovich, D. K. Mulligan, and S. B. Wicker. Inferring personal information from demand-response systems. *IEEE Security and Privacy*, 8(1):11–20, 2010.
25. P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7:75–77, 2009.
26. T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – Proceedings of CRYPTO*, 1991.
27. R. Petric. A privacy-preserving concept for smart grids. In *Sicherheit in vernetzten Systemen: 18. DFN Workshop*, pages B1–B14. Books on Demand GmbH, 2010.
28. A. Rial and G. Danezis. Privacy-preserving smart metering. Technical report, Microsoft Research, November 2010.

29. F. Sultanem. Using appliance signatures for monitoring residential loads at meter panel level. *Power Delivery, IEEE Transactions on*, 6(4):1380–1385, oct 1991.
30. B. Thompson, S. Haber, W. G. Horne, T. Sander, and D. Yao. Privacy-preserving computation and verification of aggregate queries on outsourced databases. In *Privacy Enhancing Technologies*, pages 185–201, 2009.
31. M. Torchia. Idc marketscape: North american ami communication network 2011 vendor assessment.
32. J. Vaidya, Y. M. Zhu, and C. W. Clifton. *Privacy Preserving Data Mining (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.