Error-resilient Masking Approaches for Privacy Preserving Data Aggregation

Fabian Knirsch, *Student Member, IEEE,* Günther Eibl, *Member, IEEE,* and Dominik Engel, *Member, IEEE*

***Abstract*—The deployment of future energy systems promises a number of advantages for a more stable and reliable grid as well as for a sustainable usage of energy resources. The efficiency and effectiveness of such smart grids rely on customer consumption data that is collected, processed and analyzed. This data is used for billing, monitoring and prediction. However, this implies privacy threats. Approaches exist that aim to either encrypt data in certain ways, to reduce the resolution of data or to mask data in a way so that an individuals' contribution is untraceable. While the latter is an effective way for protecting customer privacy when aggregating over space or time, one of the drawbacks of these approaches is the limitation or full negligence of device failures. In this paper we therefore propose a masking approach for spatio-temporal aggregation of time series for protecting individual privacy while still providing sufficient error-resilience and reliability.**

***Index Terms*—Privacy, smart metering, masking, fault tolerance**

I. Introduction

The movement towards intelligent and integrated future energy systems (smart grids) promises a more stable and reliable grid as well as the integration of renewable energy resources. However, this requires the processing and analysis of detailed data from a number of decentralized entities. Typical applications are billing, demand response and network monitoring. Data granularity needs to include both, scalable spatial resolution and sufficiently high time resolution. This poses a number of threats to customer privacy [1], [2].

Recent development in the domain of the smart grid has shown the need for reliable, secure and privacy-aware data collection and aggregation (see, e.g., [3]).

In order to protect customer privacy in the smart grid, approaches have been developed that obfuscate individual consumption at the household-level, e.g., battery based approaches such as [4] and [5], [6]. Another approach for obfuscating a households load curve is to adapt the load curve of existing appliances, such as electrical water heaters [7]. While these approaches protect the privacy of *single* load curves, the scope of this paper is, on the aggregation of load curves over a number of households. In contrast to these approaches, where privacy depends e.g. on the capacity of the energy storage device, masking is provably privacy-preserving [8].

When data is aggregated over a number of participants it has to be assured that (i) the data minimization principle is fulfilled, e.g., the recipient only receives the aggregate; and (ii) any other node in the network cannot gain information about anyone's contribution. Similarly, when data is aggregated over time, the receiver should only receive the sum for the previously agreed time slot. There are a number of approaches that aim at increasing privacy in smart metering. Related work in this domain mainly addresses (i) homomorphic encryption; (ii) masking; (iii) differential privacy; and (iv) multiple resolutions. Schemes and protocols that draw on homomorphic encryption (e.g., [9] and [10]) allow to process and aggregate data without access to the plaintext. Most of these approaches apply the Paillier crypto scheme [11] and therefore each smart meter uses the aggregator's public key to encrypt. Encrypted values may be sent in a hop-by-hop manner to all smart meters and finally to the aggregator. While homomorphic encryption is a reliable and powerful method for aggregating values, one of the drawbacks is its moderate efficiency in terms of computational complexity and data expansion, such as for the Paillier crypto scheme, where a plaintext in $\mathbb{Z}_n^*$ expands to a cyphertext in $\mathbb{Z}_{n^2}^*$ [11]. Another approach for spatio-temporal aggregation of power consumptions is presented in [9]. This approach combines the Paillier crypto scheme, which is used for its homomorphic property, and random shares. Although all participants possess the private key, decryption is prevented by using random shares in the exponent. In case of a faulty smart meter in the protocol an additional round is necessary.

Differential privacy is given if – with high probability – it cannot be seen from the aggregated values whether an individual participates in a database or not. Therefore an individual's participation only reveals limited personal information. While differential privacy itself is error-resilient and can be used alone [12], it is often combined with other methods that are not error-resilient, such as masking, e.g., [13] and [14]. Since differential privacy is achieved by the addition of noise, the resulting aggregate is not exact. As a consequence, the utility of the aggregated load curve decreases.

In order to achieve fault tolerance, Chan et al. [15] extend the protocol of [14], by organizing user groups in a tree. Since each user is in $\log n$ user groups, fault tolerance increases total communication from $O(n)$ to $O(n \log n)$.

Multi-resolution approaches such as [16] and [17] propose to increase privacy by splitting a load curve into a number of time-domain resolutions that are distributed to different recipients. While this is an effective method for not revealing details of an individual's load curve, it is not feasible for applications that need data in real-time.

Finally, there are a number of approaches suggesting the masking of meter values for a privacy-aware aggregation

Fabian Knirsch, Günther Eibl and Dominik Engel are with the Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, Urstein Sued 1, A–5412 Puch/Salzburg, Austria, e-mail: {fabian.knirsch, guenther.eibl, dominik.engel}@en-trust.at, Fabian Knirsch is with the Department of Computer Sciences, University of Salzburg, Jakob-Haringer-Str. 2, 5020–Salzburg, Austria

protocol. In [18], Kursawe et al. present a set of four protocols and distinguish between aggregation protocols and comparison protocols. The former are designed to compute the aggregated sum, whereas the latter require that the aggregator already knows the approximate sum. The proposed low-overhead protocol is extremely efficient in terms of computational complexity and communication needs, however it lacks the ability to deal with smart meter outages. This means that the final aggregate is invalid, if one of the key shares is missing. In order to retrieve a valid sum the aggregator has to start over the aggregation process.

In [19], Marmol et al. present a privacy enhancing aggregation protocol based on the Castellucia-Mykletun-Tsudik encryption scheme, which in order to be secure, requires that keys are not reused. The aggregation protocol therefore uses a ring-based topology that sequentially updates the smart meter keys before masking each values and the aggregator decrypts with a single static key. The protocol can be extended in order to deal with faulty smart meters, however this poses additional overhead and is not covered by the basic protocol.

In [14], Shi et al. propose a method that combines masking and distributed differential privacy. Each participating meter masks its reading with noise and the aggregator is able to finally compute a noisy version of the desired statistics, e.g., the summation of values. This method (by design) does not take into account the failure of participants, as privacy cannot be guaranteed for partial decryption.

In [20], Danezis et al. present an approach which is also based on masking. In contrast to this paper, their aim is to evaluate complex functions on one or more smart meters' values by splitting them up into Boolean circuits. However, this comes at the cost of more computing rounds, negatively affecting bandwidth and latency.

In summary, multiple resolutions are not suitable for real-time applications, homomorphic encryption is prone to high computational complexity on the smart meter side and differential privacy does not yield the correct sum. While there are promising approaches for spatial or temporal aggregation that protect customer privacy by using a masking approach, state of the art protocols exhibit none or only limited resilience for smart meter failures. Table I compares related work to our approach with respect to the ability for achieving the exact aggregation result, spatial and/or temporal aggregation, error-resilience and the requirement for a trusted third party (TTP), e.g., for key distribution or authentication. Categories denoted in the table are homomorphic encryption (HE), differential privacy (DP), masking (M) multi-resolution (MR) and Boolean circuits (BC). Half of the approaches are not error-resilient or at least error resilience is not discussed. For the remaining error resilient approaches, the overhead is categorized as negligible or not negligible. Negligible overhead consists of awaiting timeouts or rerouting requests for some smart meters along a line. The overhead is not negligible if any of the following is needed: additional rounds ([10], [13]), additional parties (in [9] the manufacturer is contacted) and additional messages to or from each smart meter involved in the protocol ([19]). Note that in the billing protocol of [12], the exact result is not provided immediately, but customers might pay more than actually needed and get a deposit on that amount. Compared to other approaches, our protocol contributes to the state of the art as it yields the exact result, is error-resilient, is suitable for spatial and temporal masking at negligible overhead, does not rely on a TTP for privacy and has a low complexity compared to approaches using homomorphic encryption.

In this paper, we present a privacy-aware approach for spatio-temporal aggregation of time series data. We apply a masking scheme that obfuscates individual contributions and yields the correct result upon aggregation. Since a single, invalid random share can have a devastating effect on the aggregate, the proper use of random shares is checked using homomorphic hashes. We do not restrict ourselves to a concrete operation, however we will prove correctness for summation, which is the most common aggregation. While this approach can be applied to any time series data, we will focus on use cases with smart meter readings for power consumption. In addition, we focus on a high degree of error-resilience which is crucial in terms of the distributed nature of the smart grid. If one or more smart meters fail during the aggregation process, the protocol is capable of providing an accurate aggregation at the same level of privacy. The term *error resilience* in this paper explicitly refers to the outage of a smart meter or the malfunctioning of communication links and not to an outage of the data concentrator.

The rest of the paper is structured as follows: Section II motivates the use of a more error-resilient aggregation protocol and introduces the notation. Section III introduces the preliminaries for this paper, including the masking scheme and homomorphic hashes. Section IV discusses the proposed protocol for token-based sequential masking. In Section V the scheme is analyzed with respect to adversarial models and attack scenarios. Section VI describes the prototypical implementation and Section VII summarizes this work and gives an outlook to future research.

II. Application Scenario

This section describes the problem domain and motivates spatial, temporal and spatio-temporal aggregation of time series in the smart grid. Further, the general masking approach and the underlying privacy considerations are introduced.

A. Use Case Description

Aggregating data in the smart grid is a basis for many use cases. In addition, aggregating data is a method for protecting customers' privacy. In [9], Erkin and Tsudik, in [21], Jawurek et al. and in [22] McKenna et al. propose typical use cases that pose a need for privacy-preserving spatial and temporal aggregation:

- **Network Stability and Monitoring.** The stability of the power grid is maintained by network operators and utilities by collecting high-frequency measurements for voltage levels, phase shifts and power consumptions. While this application requires a high temporal resolution, privacy can be protected by aggregating over a number of households, e.g., over those connected to the same substation.

TABLE I
COMPARISON OF RELATED WORK. METHODS ARE HOMOMORPHIC ENCRYPTION (HE), DIFFERENTIAL PRIVACY (DP), MASKING (M), MULTI-RESOLUTION (MR) AND BOOLEAN CIRCUITS (BC). OUR APPROACH GIVES THE EXACT RESULT FOR BOTH, SPATIAL AND TEMPORAL AGGREGATION AT NEGLIGIBLE OVERHEAD EVEN IF ONE OR MORE SMART METERS FAIL. A TRUSTED THIRD PARTY (TTP) IS REQUIRED FOR AUTHENTICATION PURPOSES ONLY.

Approach	Methods	Exact result	Spatial	Temporal	Error Resilience	TTP
Erkin et al. [9]	HE	✓	✓	✓	✓	key distr.
Li et al. [10]	HE	✓	✓	–	✓	key distr.
Danezis et al. [12]	DP	–	–	✓	–	key mgmt.
Chan et al. [15]	HE, DP	–	✓	–	✓	key distr.
Acs et al. [13]	M, DP	–	✓	–	✓	key distr.
Shi et al. [14]	HE, M, DP	–	✓	–	–	key distr.
Efthymiou et al. [16]	MR	✓	–	✓	–	key distr.
Engel et al. [17]	HE, MR	✓	✓	✓	–	key distr.
Kursawe et al. [18]	M	✓	✓	–	–	share distr.
Marmol et al. [19]	HE, M	✓	✓	–	✓	key mgmt.
Our approach	M	✓	✓	✓	✓	authentication

- **Settlement and Profiling.** Energy providers trade at a wholesale market which requires them to have detailed information about the current energy demand at a high temporal resolution. Similarly to network monitoring, privacy can be protected by aggregating over a number of households.
- **Billing.** Billing is a transaction between the customer (smart meter) and the utility. For billing, spatial aggregation is not applicable. Time-of use pricing with fixed rates where the price is piecewise constant can be handled by temporal aggregation over the corresponding time intervals.

In a practical setting, smart meters are connected to a data concentrator or aggregator. Depending on the use case the smart meter will participate in either a spatial, temporal or spatio-temporal aggregation protocol. Basic temporal aggregation can be achieved in practice by simply holding back values for a certain period of time and then submitting the sum of these values at once. The protocol presented in this paper, is designed for efficient, error-resilient spatial aggregation, however, also allows for temporal masking at negligible additional computational and communication cost.

B. Problem Statement and Notation

The system consists of the following actors: a single data concentrator DC and a set of participating smart meters $\mathrm{SM}_1, \ldots, \mathrm{SM}_N$. Each smart meter SM_i measures a time series of values, i.e., at time t it measures $m_{i,t}$. Each measurement $m_{i,t}$ is a scalar integer value in the range $\{0, \ldots, c-1\}$, e.g., in practice c could be chosen as 2^{16}. The goal is to provide the aggregate $A_t = \sum_i m_{i,t}$ to the data concentrator without revealing the load profiles of the individual smart meters. N should be large enough for successfully hiding individual load profiles in the aggregated load profile. Later we will show that the protocol is also capable of providing a temporal aggregation for each individual smart meter to DC with negligible overhead. Even more, arbitrary subsets of aggregations over space and time can be calculated.

1) Spatial Aggregation: Spatial aggregation is any aggregation over a set of smart meters $G = \{\mathrm{SM}_i;\ i = 1, \ldots, N\}$ at a fixed time t, i.e., the calculation of $A_t = \sum_i m_{i,t}$. This is typically applied for calculating the total consumption at a certain point in time for a defined neighborhood. Such aggregations are needed for network monitoring and load balancing. In the following it will be convenient to use another representation for G as a tuple $G = (\mathrm{ID}_1, \ldots, \mathrm{ID}_N)$ that contains the IDs of the smart meters participating in spatial aggregation. The desired order of the smart meters can be easily saved in the *sending list* $L = (\mathrm{ID}_{\mathrm{DC}}, G, \mathrm{ID}_{\mathrm{DC}}) = (\mathrm{ID}_{\mathrm{DC}}, \mathrm{ID}_1, \ldots, \mathrm{ID}_N, \mathrm{ID}_{\mathrm{DC}})$ which contains the ordered IDs of the smart meters of the aggregation group with the ID of the DC prepended and appended.

2) Temporal Aggregation: Temporal aggregation is any aggregation over a (sequential) series of values $(m_{i,t})_{t=1,\ldots,T}$ for a fixed smart meter SM_i, i.e., the calculation of $A_i = \sum_{t=1,\ldots,T} m_{i,t}$. This is typically applied for billing, where the total energy consumption in a given period of time is of interest for the aggregator. For large billing periods T privacy is preserved, whereas for small billing periods privacy is not guaranteed. This issue of the impact of time series resolution on privacy is discussed in [1] and [2].

3) Spatio-Temporal Aggregation: The protocol that is proposed in this paper is particularly designed for spatio-temporal aggregation, hence given a set of smart meters and a series of values for each smart meter, arbitrary sets of aggregations over space G and time intervals $[t_0, t_1]$ can be calculated, i.e., $A_{G,[t_0,t_1]} = \sum_{i \in G} \sum_{t \in [t_0,t_1]} m_{i,t}$. Figure 1 illustrates the two dimensional space and an arbitrary spatio-temporal aggregation over a group G, which is represented as a rectangle.

We assume that smart meters are able to bidirectionally communicate with the DC and with each other. Therefore it is likely, that in practice aggregation groups are restricted to contain only smart meters that can communicate with each other.

III. PRELIMINARIES

This section describes the preliminaries for the proposed protocol. The protocol builds on masking, a lightweight and

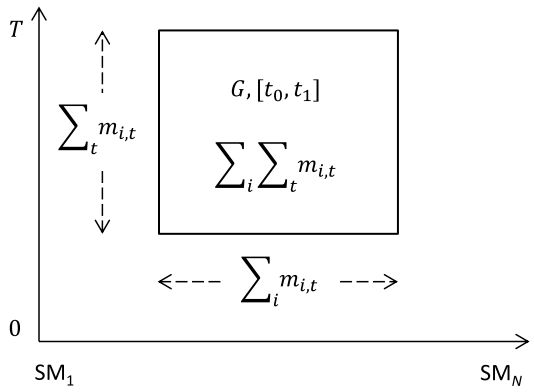


Fig. 1. Illustration of spatio-temporal aggregation over a group G as a rectangle in the two dimensional space.

established scheme for hiding individual contributions in aggregation protocols. Further, the protocol uses homomorphic hashes that allow the aggregator to check the correctness of shares.

A. Masking Approach

For masking, a value $m_{i,t}$, is combined with a random share $s_{i,t}$ in the range $\{0, \ldots, k-1\}$ resulting in a masked value

$$\tilde{m}_{i,t} = m_{i,t} + s_{i,t} \mod k, \tag{1}$$

where the modulus k must be larger than the highest possible aggregation value. If a cryptographic pseudo random number generator exploits the full range of the plain value, masking is secure [8]. Therefore, k could be chosen as $k = TNc \geq \sum_{t=1\ldots T} \sum_{i=1\ldots N} m_{i,t}$ in the most general case of spatio-temporal aggregation, where c is the upper bound of the range for the measurement values. Each smart meter sends the masked value $\tilde{m}_{i,t}$ to the DC which calculates their sum $\sum_{i=1\ldots N} \tilde{m}_{i,t}$. Due to the masking, an individual load profile has all properties of a random number time series and therefore reveals no information to the DC. However, at each time t, typically the random shares $s_{i,t}$ of the N smart meters are created in a way that they cancel each other out when the sum is formed, i.e., $\sum_{i=1\ldots N} s_{i,t} = 0$. As a consequence, the spatially aggregated masked values $\sum_{i=1\ldots N} \tilde{m}_{i,t}$ are equal to the desired sum of spatially aggregated measurements $\sum_{i=1\ldots N} m_{i,t}$. Masking for smart grid applications is proposed by e.g., [18], [19].

B. Homomorphic Hash

An additive homomorphic hash function H is a hash function with the property that for all x and y the equality $H(x+y) = H(x)H(y)$ holds (note that here the group operation in the output domain is arbitrary, but usually a product). Such a hash function $H : \mathbb{Z}_p \mapsto \mathbb{Z}_l$ can, e.g., be constructed around the discrete-log assumption by a generator g as $H(v) = g^v \mod l$, which yields a collision-resistant hash as discussed in [23]. The homomorphic property can easily be verified by calculating

$$\begin{aligned} H(v_1)H(v_2) &= g^{v_1} g^{v_2} = g^{v_1+v_2} \\ &= H(v_1+v_2) \pmod l. \end{aligned} \tag{2}$$

Choosing a suitable generator is also discussed in literature, e.g., [24]. In Section VI, p is chosen as $p = 2^{16}$ and l is suggested to be set to $l = 2^{256}$ for a good privacy-efficiency tradeoff.

IV. Privacy-Preserving Protocol

In this section, our privacy-preserving protocol is developed step by step. First, a basic protocol for spatial aggregation is explained which is subsequently expanded in order to increase fault-tolerance, enable spatio-temporal aggregation and include principal verification elements.

A. Privacy for Spatial Aggregation

In this section a basic spatial aggregation algorithm is presented which still suffers from some flaws that will later be eliminated by the actual algorithm. This algorithm is not yet capable of handling failures, but introduces the principal masking scheme. In this protocol the masked values are directly sent to the DC. In contrast to other protocols, here the shares $s_{i,t}$ are created independently from each other by sampling from $\{1, \ldots, k-1\}$. As a consequence, they do not sum to zero. Here, their sum is calculated using the ring part of the topology (Figure 2), i.e., the random shares are sent and summed up *between the smart meters* of the aggregation group G. Then, the obtained aggregated sum of shares is subtracted from the sum of the masked measurements yielding the desired sum of measurements $\sum_{i=1\ldots N} m_{i,t}$. The process for one round is shown in Figure 2. For sake of readability the time index t is often omitted and the notation $S_i = \sum_{j=1\ldots i} s_{j,t}$ is introduced. The DC sends an initial random share $S_0 = s_0$ to the first smart meter SM_1 which creates its own random share s_1. This share is added to its measurement value m_1 yielding $\tilde{m}_1$ which is sent directly to the DC. Additionally, the smart meter adds up the two shares calculating $S_1 = s_0 + s_1$ which it sends to node SM_2. This continues up to the last node N which calculates

$$S_N = S_{N-1} + s_n = S_{N-2} + s_{n-1} + s_n = \ldots = \sum_{i=1}^{N} s_i$$

and sends it to the DC. Finally, DC calculates $\sum_i \tilde{m}_i - S_N = \sum_i m_i$ which is the desired aggregated load. The corresponding algorithm is shown in Algorithm 1.

As already reasoned above, the data concentrator retrieves the correctly aggregated value. Privacy is preserved because DC only gets masked values. Since the DC only gets the sum of the shares, only the aggregated sum of the measurements can be recovered. The measurement of a smart meter is also hidden from other smart meters because only the shares are sent between them. The DC can itself add a random share s_0 in order to increase the difficulty for obtaining the value of SM_1. Thus, for spatial aggregation, the logic topology for the masking algorithm is a combination of a star and ring topology (Figure 2). All smart meters are connected to the DC in a star-shaped topology. These links are used for submitting masked values to the DC. In addition, the DC and all smart meters are connected in a ring-shaped topology, with each actor having a designated predecessor and a designated successor.

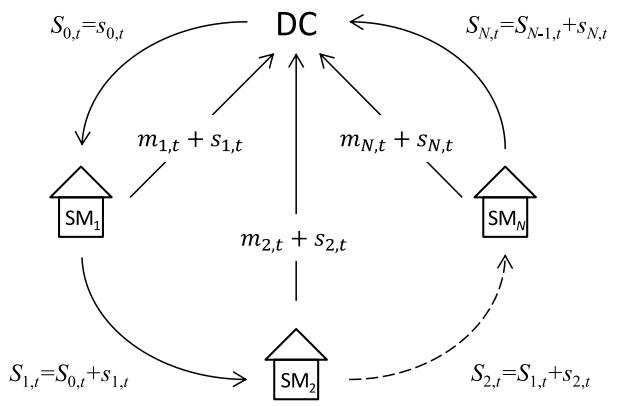


Fig. 2. Error-resilient spatial masking scheme for one round. The data concentrator (DC) triggers the token-based sequential reading. Each SM_i calculates a share, submits the masked value to DC and submits its share to the successive smart meter.

Algorithm 1: Preliminary spatial aggregation algorithm.

Initialization
Provide sending list L to all smart meters and to DC

One round of reading t
All participating parties i generate a random share $s_{i,t}$
$p = 0$
DC sends $M_{\mathrm{DC}\to 1} = (S_0||\mathrm{ID}_0) = (s_{0,t}||\mathrm{ID}_{\mathrm{DC}})$ to SM_1
for $i \leftarrow 1$ **to** N **do**
  $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} \mod k$
  SM_i sends $M_{i\to\mathrm{DC}} = \tilde{m}_{i,t}$ directly to DC
  $S_i = S_p + s_{i,t} \mod k$
  SM_i sends $M_{i\to i+1} = S_i$ to $L(i+1)$
  $p = i$
end

Aggregation
DC calculates $A_t = \sum_i \tilde{m}_{i,t} - S_N$

B. Fault Tolerance

In order to obtain the correct sum of the measurements it is crucial that the sum of the shares cancel each other out, as in our case, where S_N exactly equals $\sum_i s_i$ which is subtracted at the end. As a consequence, even one wrong share can destroy the correctness of the result. For some privacy preserving methods that combine encryption with masking even a small error can have the devastating effect that the resulting aggregate cannot be decrypted at all. For the basic algorithm above, the effect of a wrong share would be an incorrect aggregation result. In this section we modify the basic algorithm in order to improve the fault tolerance.

There are four cases where a fault can occur:

(i) Failure of smart meter at initialization;
(ii) Failure of connection between smart meter and DC;
(iii) Failure of connection between smart meters; and
(iv) Failure of smart meter during normal operation.

If a smart meter does not respond in the initialization phase (i), it is not a member of the sending list and the smart meter is not included in the aggregation process at all. The system may attempt a new initialization after a certain amount of time or once the smart meter is available.

In the case that only the connection between a smart meter and the DC fails (ii), the value $\tilde{m}_{i,t}$ is not available, which enables the DC to detect the communication failure with the smart meter. However, the connection to other smart meters still works and consequently the shares of the faulty smart meter will be part of the sum of shares S_N. As a consequence, the aggregation result cannot be corrected by the DC because it does not have, and should not have, the value of the share of the faulty smart meter $s_{i,t}$.

If the connection between smart meters $i-1$ and i fails (iii), smart meter $i-1$ eventually realizes that its message has not reached smart meter i, depending on the protocol and the type of connection failure. Still, the basic algorithm fails, since it allows smart meter $i-1$ only to send its value S_{i-1} to the next smart meter in the sending list.

If a smart meter fails during normal operation (iv), in terms of communication errors this fault can be viewed and treated as the combination of the preceding two faults.

The solution to these problem consists of: (i) providing a means for the preceding smart meters to detect the failure of the connection to its faulty neighbor. This can be achieved by requiring the sending of an acknowledgment (Ack) signal of a smart meter to its predecessor; and (ii) providing a way to skip a faulty smart meter in the ring part where the shares are summed up and in the calculation of the sum of the masked measurement at the DC. This is achieved by selecting the next smart meter from the sending list. Algorithm 2 is an extension of the preliminary algorithm that handles these cases.

Now that the algorithm is resistant to failures of smart meters and connections, the algorithm will later also be made more resistant with respect to failures occurring in sent shares (Section IV-D).

Algorithm 2: Fault tolerant spatial aggregation

Initialization
Provide sending list L to all smart meters and to DC

One round of reading t
All participating parties i generate a random share $s_{i,t}$
$i = 1$, $p = 0$, $n = 2$
DC sends $M_{\mathrm{DC}\to 1} = (S_0||\mathrm{ID}_0) = (s_{0,t}||\mathrm{ID}_{\mathrm{DC}})$ to SM_1
while $i \leq N$ **do**
  SM_i checks that the received ID is a predecessor of SM_i in L
  SM_i sends an Ack-signal $M^{\mathrm{Ack}}_{i\to p}$ to $L(p)$
  $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} \mod k$
  SM_i sends $M_{i\to\mathrm{DC}} = \tilde{m}_{i,t}$ directly to DC
  $S_i = S_p + s_{i,t} \mod k$
  SM_i sends $M_{i\to n} = S_i$ to $L(n)$.
  $p = i$
  if SM_i gets an Ack-signal within Δt **then**
    $i = n$
  else
    $n = n + 1$
    send S_i to $L(n)$
  end
end

Aggregation
DC calculates $A_t = \sum_i \tilde{m}_{i,t} - S_N$

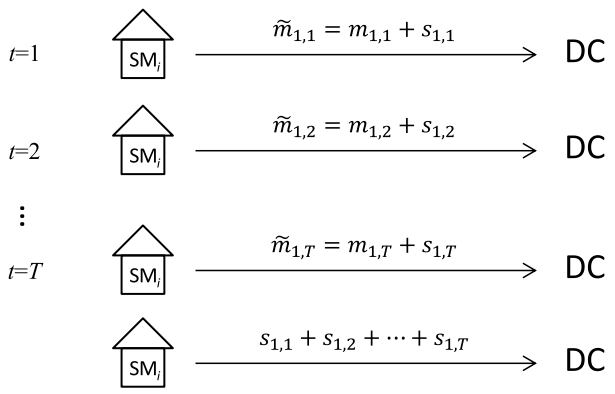


Fig. 3. Temporal aggregation masking scheme for one smart meter. The smart meter (SM_i) sends masked values to the data concentrator DC. At the end of period T, the sum of shares is sent to the aggregator.

C. Spatio-Temporal Aggregation

In this section, the spatio-temporal aggregation is outlined. Since temporal aggregation is performed separately for each smart meter, only the star topology is required (Figure 3). Temporal masking and aggregation can be done in a straightforward way, (Algorithm 3), with $S_T = \sum_{\tilde{t}=0}^{T} s_{i,\tilde{t}}$. For sake of readability the smart meter index i is omitted.

With both, spatial and temporal aggregation available, spatio-temporal aggregation $A_{G^*,[t_0,t_1]}$ – where G^* denotes an arbitrary subset of smart meters and $[t_0, t_1]$ denotes the time period of interest – can easily be obtained with only little overhead by calculating

$$A_{G^*,[t_0,t_1]} := \sum_{i\in G^*} \sum_{t\in[t_0,t_1]} m_{i,t} \quad (3)$$

$$= \sum_{t\in[t_0,t_1]} \sum_{i\in G} m_{i,t} - \sum_{i\notin G^*} \sum_{t\in[t_0,t_1]} m_{i,t} \quad (4)$$

$$= \sum_{t\in[t_0\ldots t_1]} A_t - \sum_{i\notin G^*} A_i. \quad (5)$$

The whole algorithm (Algorithm 4) is subject to some practical restrictions. In the algorithm, spatial aggregation is performed over all smart meters (aggregation group G) connected to a data concentrator. Spatio-temporal aggregation can then be done over arbitrary spatial subgroups ($G^* \in G, [t_0, t_1]$) in the last step, but with the limitation of a pre-specified time interval, which must be the same for all smart meters. For sake of readability, the steps for calculating S_T as shown in Algorithm 3 are omitted. Note that choosing the appropriate (sub-)groups is relevant for ensuring privacy, in the sense that a single smart meter's value cannot be recovered by combining aggregated values, e.g., when two groups are identical except for a single smart meter. Choosing more spatial groups over which to aggregate for a *specific time* would result in an increase of spatially aggregated shares (one value per group) on the ring part of the topology. The number of values to be transferred between smart meters and the DC on the star part of the topology would not be influenced.

For the algorithm as stated here, the aggregation period must be known in advance. Alternatively, the smart meter could internally save its random shares and deliver the sum of the shares needed for temporal aggregation on request, providing more flexibility in choosing the aggregation period $[t_0, t_1]$. In any case, while the smart meter has nearly no control over the spatial extent of aggregation, it has full control regarding purely temporal aggregation allowing, e.g., only temporal aggregation over periods that are longer than a user-specific "privacy-safe" duration.

Algorithm 3: Temporal Aggregation

Reading of T values
$S_0 = 0$
for $t \leftarrow 0$ **to** T **do**
  Generate a random share $s_{i,t}$
  $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} \mod k$
  $S_t = S_{t-1} + s_{i,t}$
  Send $M^t_{i\to\mathrm{DC}} := \tilde{m}_{i,t}$ to DC
end
Send $M_{i\to\mathrm{DC}} := S_T$ to DC
Aggregation DC calculates $A_i = \sum_t \tilde{m}_{i,t} - S_T$

D. Correctness of Shares

While the algorithm so far is fault-tolerant with respect to complete failures of communication links or smart meters, still a single, wrong share $s_{i,t}$ would suffice to produce a wrong aggregation result. Consequently, the network operator is likely to be interested in also having some insight, especially into the correctness of the shares sent between the smart meters. For the correctness of the aggregated spatial value, it is necessary that the value for the sum of shares provided to the DC, S_N, is indeed the sum of the shares used by the individual smart meters in the calculation of the masked value. More precisely, a check that (i) $S_N = \sum_i s_{i,t}$ and a check that (ii) $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} + s^0_i$ is desired. Additionally, a static secret s^0_i shared between the DC and each SM_i is introduced in order to further increase privacy for intercepted messages as discussed in Section V.

Since both checks are additions of values, they can be performed by applying additively homomorphic hash functions to the messages. The properties of such a hash function are described in the preliminaries in Section III.

In the algorithm, for each $\tilde{m}_{i,t}$, SM_i additionally sends the homomorphic hashes $H(m_{i,t})$ and $H(s_{i,t})$ to the DC. Given the homomorphic property of H, DC can check that no share has been altered through comparing

$$H(S_N) = \prod_i H(s_{i,t}) \quad (6)$$

before aggregating; and DC can check that $s_{i,t}$ has been used for calculating $\tilde{m}_{i,t}$ by comparing

$$H(\tilde{m}_{i,t}) = H(m_{i,t})H(s_{i,t})H(s^0_i). \quad (7)$$

The static secret share s^0_i has been provided to each smart meter upon initialization.

In order to enable a fast correction of errors of shares, a method is needed that pinpoints the place of the error in case that check (6) shows a difference between the shares. The

Algorithm 4: Final aggregation algorithm

Initialization
Provide sending list L to all smart meters and to DC
$\forall i$: DC sends static secret share s_i^0 to SM_i, $S^0 = \sum_i s_i^0$

while $t \leq T$ **do**
  For all i: A_i = 0
  One round of reading t
  All participating parties i generate a random share $s_{i,t}$
  $n = 1$
  DC sends $M_{\mathrm{DC}\to 1} = (S_0||\mathrm{ID}_0) = (s_{0,t}||\mathrm{ID}_{\mathrm{DC}})$ to SM_1
  while
  DC does not get $M_{n\to 0}^{\mathrm{Ack}}$ from SM_n within Δt **do**
    $n = n + 1$
    DC sends $M_{\mathrm{DC}\to 1}$ to SM_n.
  end
  $i = n$, $p = 0$, $n = i + 1$
  while $i \leq N$ **do**
    /* Only SM_i acts */
    Parse $M_{p\to i}$ and check that ID_p is a predecessor of SM_i in L
    Send an Ack-signal $M_{i\to p}^{\mathrm{Ack}}$ to $L(p)$
    $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} + s_i^0 \mod k$
    $S_i = S_p + s_{i,t} \mod k$
    $A_i = A_i + m_{i,t}$
    Send $M_{i\to \mathrm{DC}} := (\tilde{m}_{i,t}||H(m_{i,t})||H(s_{i,t})||t)$ to DC
    Send $M_{i\to n} = S_i$ to $L(n)$.
    while SM_i does not get $M_{n\to i}^{\mathrm{Ack}}$ within Δt **do**
      $n = n + 1$
      send S_i to $L(n)$
    end
    $p = i$, $i = n$
  end
  Aggregation
  /* Only DC acts */
  Check if $H(S_N) = \prod_i H(s_{i,t})$
  For all $\tilde{m}_{i,t}$: check $H(\tilde{m}_{i,t}) = H(m_{i,t})H(s_{i,t})H(s_i^0)$
  Calculate spatial aggregate $A_t = \sum_i \tilde{m}_{i,t} - S_T - S^0$
  Increase t
end
DC calculates spatio-temporal $A_{G,[0,T]} = \sum_t A_t - \sum_{i\notin G} A_i$

idea for locating the difference is to perform one additional round of aggregation with the difference that, instead of the masked measurement $\tilde{m}_{i,t}$, the aggregated share obtained from the previous smart meter S_p is sent to the DC. The correct use of share $s_{i,t}$ can be checked as follows: the DC obtains $\tilde{m}_{i,t} = S_p$ and $H(s_{i,t})$ from SM_i and $\tilde{m}_{n,t} = S_i$ from SM_n. If the same share $s_{i,t}$ is not used for both, for the message to the DC and the message to the next smart meter, the equality

$$H(S_i - S_p) = H(s_{i,t}) \tag{8}$$

should not hold, indicating a problem between SM_p and SM_i. Due to check (7), a problem between SM_i and the DC can be identified directly, as the smart meters are assumed to act honestly. In the purely temporal scenario where the primary use case is billing, the user is as well likely to be interested in a check that the DC obtained the correct aggregated value. At the end of the billing scenario, the DC could provide $H(A_i)$ which can be compared to $\sum_t H(m_{i,t})$ by the smart meter.

It should be emphasized that these hashes are intended as checks for the correctness of data and calculations. They are neither intended to proof data integrity and authenticity nor are they intended as a proof for correctness of the calculation (also compare with Section V).

V. Security and Privacy Analysis

This section conducts a security and privacy analysis of the proposed protocol.

A. General assumptions

We generally assume all devices to be tamper proof and that attackers do not have physical access, thus the meter value itself cannot be manipulated. We further assume that communication is handled over a secure channel using state-of-the art symmetric encryption, e.g., AES [25] and some sort of authentication to tackle man-in-the-middle attacks, e.g., X.509 certificates [26].

B. Privacy-preserving property of masking

Masking is proposed by many authors as a method for the privacy-preserving aggregation of data in the smart grid (e.g., [13], [14], [18], [19]). For masking a value, the value is added to a random share modulo the upper bound of the range of the aggregate. The random shares (e.g., generated by a cryptographic pseudo random number generator) have to fully exploit the range of the possible values. The computational security of such a scheme is shown in [8]. Note that for perfect secrecy all the properties of the one-time pad would apply, i.e., key length equal to the length of the plaintext, the source of randomness and the freshness of random numbers are crucial.

C. Adversaries and privacy breaks

This protocol involves two different parties that can act as potential adversaries, smart meters and the data concentrator. In addition, for the malicious adversary model an external adversary as well as a covert adversary are considered.

A privacy break in this protocol occurs if (i) the data concentrator learns anything beyond the sending list, the spatial aggregate and the temporal aggregate; if (ii) any of the smart meters learns anything except for the sending list; and if (iii) any of the participants can tamper with the aggregate, i.e., manipulate contributions such that the aggregate becomes void. It is of particular interest that no party learns a single measurement value $m_{i,t}$.

D. Single honest-but-curious participants

In this scenario, all participants follow the proposed protocol but a single party may attempt to gain additional information. Therefore a single smart meter or the data concentrator is honest-but-curious (semi honest).

A smart meter SM_i receives from another smart meter, its predecessor, only the value S_{i-1}. Since this value is only the sum of the previous shares, SM_i cannot infer any information about any measurement values from this sum. The data concentrator DC is not able to infer either a subset of

values or a single value $m_{i,t}$ of one smart meter SM_i at one point in time t. This is achieved by masking each smart meters value with a random share $s_{i,t}$ in

$$\tilde{m}_{i,t} = m_{i,t} + s_{i,t} + s_i^0 \mod k. \quad (9)$$

The random share $s_{i,t}$ is sampled at the smart meter and the meter value $m_{i,t}$ is never released as a plain value and therefore remains untraceable. SM_i forwards its share $s_{i,t}$ combined with the share from its predecessor as $S_i = S_{i-1} + s_{i,t} \mod k$. The DC only gets S_N from which it cannot infer $s_{i,t}$.

In order to ensure the correctness of shares, each SM_i does not only submit $\tilde{m}_{i,t}$ to DC, but also $H(m_{i,t})$ and $H(s_{i,t})$, where $H(\cdot)$ is a collision-resistant homomorphic hash function constructed around the discrete-log assumption. Since the same hash value of the secret share $s_{i,t}$ is used in the checks that $S_i = S_{i-1} + s_{i,t} \mod k$ and that $\tilde{m}_{i,t} = m_{i,t} + s_{i,t} + s_i^0 \mod k$, it is ensured that the same share is used in both calculations. If one or more smart meters fail, the aggregator still receives the exact sum of the remaining smart meters and the aggregator still does not learn anything beyond that aggregate.

The length of the time interval as well as the group size will have significant impact on the privacy. However, the check if these sizes are sufficiently large can be done by the smart meter itself too, i.e., if the requested time interval or the sending list, are to short, the smart meters can decline the submission or forwarding of a value. Note that due to many failing smart meters the group size could drop below a privacy-preserving level. However, the algorithm could easily be expanded to monitor the number of remaining active smart meters and therefore refuse participation in this case. Finding suitable group sizes and time intervals in order to guarantee a certain level of privacy for practical smart grid applications is still an open research question.

E. Collusion of honest-but-curious participants

Collusions of honest-but-curious participants can occur in two different ways (i) either a subset of smart meters can collude; or (ii) one or more smart meters can collude with the data concentrator.

In the first case, a subset of smart meters collude in order to get information about some other SM_i. SM_i sends $S_i = S_{i-1} + s_{i,t}$ to its subsequent smart meter SM_{i+1}. If SM_{i+1} and SM_{i-1}, owning S_{i-1}, are part of the collusion, they can easily reconstruct the secret shares of SM_i by calculating $s_{i,t} = S_i - S_{i-1}$. However, since they do neither possess any message containing SM_i's consumption value $m_{i,t}$ nor do they possess the static secret share s_i^0, the measurement value still cannot be deduced.

However, in the latter case, where DC is also part of the collusion, this changes, because $s_{i,t}$ is the information that the data concentrator (which in addition knows s_i^0) needs, in order to calculate $m_{i,t}$ from equation (9). In order to infer a single load value, the colluding set therefore needs to contain at least the DC and the two smart meters preceding and following the attacked smart meter.

F. Malicious adversaries

While homomorphic hashes were introduced in order to detect faults and enable fast error correction, they do not suffice for detecting maliciously modified values. In this case security relies on the security measures ensuring data integrity and authenticity. In order to illustrate this point we next show how much additional effort an external malicious adversary would need in order to reach his goals.

A malicious external adversary with the goal of getting a single metered value would need to break all the secure channels connecting a smart meter to other smart meters and the data concentrator. In this case it gets the same information as the collusion set above. In contrast to the collusion scenario, it doesn't know s_i^0. However, a static value only offers very limited privacy protection, e.g., typically non-intrusive load monitoring analyses [27] consider differences of subsequent values, which are not effected by a static value.

Now, a covert adversary is considered, whose goal is not only to obtain, but also to change a measurement and thereby forge the aggregation result without being detected. In this case, the adversary would need to break all the secure channels of a smart meter and additionally change the hashes in such a way that the checks of the hashes still work. Through a man-in-the-middle attack, the adversary could consistently forge a meter value $m_{i,t} \to m_{i,t} + \Delta m$ and the share $s_{i,t} \to s_{i,t} - \Delta m$. This has the effect that the unknown value $\tilde{m}_{i,t}$ stays the same. Additionally changing $H(m_{i,t}) \to H(m_{i,t})H(\Delta m)$, $H(s_{i,t}) \to H(s_{i,t})H(\Delta m)^{-1}$ and $S_i \to S_i - \Delta m$ would then ensure, that both checks preceding the aggregation step of Algorithm 4 would fail in detecting the forgery.

G. Summary

In summary, we showed the security and privacy preserving property of our protocol for single honest-but-curious participants. In case of a collusion of honest-but-curious participants, in order to attack a single smart meter, at least the data concentrator and the preceding and the following smart meter are needed. Finally, a malicious adversary would have to break all the secure communication links in order to learn measurement values or forge the aggregation result. However, given state-of-the-art cryptographic protocols and algorithms for the authentication and encryption of communication links, this is likely to be infeasible. In the case that a smart meter has been compromised, it is impossible to assure a correct reporting of measurements. While this is out of scope for this paper, smart meters could be equipped with trusted hardware (e.g., based on a Trusted Platform Module as proposed by [28]).

VI. Evaluation

In this section we evaluate the proposed masking protocol with respect to complexity, network traffic and its applicability to efficiently run on low-end devices.

A. Complexity

In terms of complexity we consider the final algorithm as listed in Algorithm 4. Table II shows a detailed analysis of the

complexity for initialization where N denotes the number of smart meters in the sending list. Each value is given as both, the total number and the corresponding complexity.

TABLE II
COMPLEXITY FOR INITIALIZATION.

	SM_i	DC
Gen. of random number		$O(1)$
Messages inbound	$O(1)$	$O(1)$
Messages outbound	$O(1)$	$O(N)$

Table III shows the complexity for one round of reading including the calculation of the spatial aggregate if no faults occur and if k smart meter fail. Without faults, all operations conducted in the smart meter are of constant time. For DC, complexity is not of constant time, but linearly growing with the number of smart meters in the group. Generally, while the smart meter has limited computational capacities, the data concentrator will be a much more powerful device, and thus the protocol scales well with the size of the grid.

TABLE III
COMPLEXITY FOR ONE ROUND OF SPATIAL AGGREGATION (WITH NO FAULTS AND k FAULTS) FOR A SINGLE SM_i AND THE DC. IN THE CASE OF FAULTS IT IS ASSUMED THAT THE k SMART METERS FOLLOWING SM_i FAIL, WHICH IS THE WORST CASE FOR SM_i. IF THE COMPLEXITY CHANGES, ENTRIES ARE MARKED BOLD.

	SM_i		DC	
	no	k	no	k
Addition	$O(1)$	$O(1)$	$O(N)$	$\boldsymbol{O(N-k)}$
Multiplication			$O(N)$	$\boldsymbol{O(N-k)}$
Random number	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Hash	$O(1)$	$O(1)$	$O(N)$	$\boldsymbol{O(N-k)}$
Messages in	$O(1)$	$O(1)$	$O(N-k)$	$O(N-k)$
Messages out	$O(1)$	$\boldsymbol{O(k)}$	$O(1)$	$O(1)$

In case faults occur, e.g., the k smart meters following SM_i are unavailable, the complexity for outbound messages for SM_i changes from $O(1)$ to $O(k)$, as the smart meter attempts to send the message to all k following smart meters in the sending list. In worst case, all smart meters following the first smart meter fail, which results in a complexity of $O(N)$. While the complexity for the smart meter increases in this case, the complexity for the data concentrator is decreased from $O(N)$ to $O(N-k)$.

Table IV shows the additional complexity if the temporal aggregate over T measurements is calculated in addition to the spatial aggregate. Although, the number of additions for DC is NT, these calculations do not need to be performed upon requesting the temporal aggregate, but can be done in each round leading to only N additions at the end of each round.

TABLE IV
ADDITIONAL COMPLEXITY FOR TEMPORAL AGGREGATION OF T MEASUREMENTS.

	SM_i	DC
Addition	$O(T)$	$O(NT)$
Messages inbound	$O(1)$	$O(N)$
Messages outbound	$O(1)$	$O(N)$

B. Implementation

This protocol has been prototypically implemented in Java (Oracle Java JDK 1.7 for ARM) and is designed to run on a Raspberry Pi 2. This small computer is a low-cost and low-power device that resembles the computational abilities of a smart meter. Our implementation will be available open source. For evaluating the prototypical implementation, two settings are prepared: (i) in our lab setting, a standard personal computer serves as the data concentrator and Raspberry Pis serve as smart meters, all connected in a LAN environment; and (ii) in our virtualized setting up to 100 smart meters are simulated in order to evaluate the behavior of the protocol if smart meters fail.

In order to account for the hot-spot property of the used Java virtual machine, i.e., the run-time optimization of frequently executed code, the first measurements are discarded. For evaluating the performance on low-power devices we investigate the following setting in our lab: we start with one data concentrator that collects spatio-temporally aggregated data and a group size of four smart meters. The generator for the homomorphic hash implementation uses safe primes with a bit length b. The bit length b has to be chosen such that $\sum_{i=1}^{N} m_i \leq \frac{2^b-1}{N}$. The random shares are less or equal than 2^{b-2}.

The methodology for measuring is as follows:

1) **Initialization**. DC and $\mathrm{SM}_1, \ldots, \mathrm{SM}_N$ are started, with $N = 4$ in the lab setting and $N = \{20, 40, 60, 80, 100\}$ in the virtualized setting. All smart meters connect to DC and report themselves as available. Finally, DC provides the sending list and other initialization material to all SM_i.
2) **Spatial Aggregation.** A controller program triggers DC to start the spatial aggregation and all SM_i report their consumption and forward their share, respectively. In our implementation a smart meter receives the share from the previous smart meter, then generates its own share, submits the masked share to the data concentrator and finally calculates and forwards the new share to its successor. Some of these operations can be parallelized or even precomputed. However, it leads to a high load on the communication link of the DC if all smart meters report their consumption at the same time.
3) **Temporal Aggregation.** For temporal aggregation the controller again triggers DC and all SM_i are requested to report their time-aggergated share S_T for revealing the temporal sum. Again, all smart meters are triggered in a sequence.

We do not consider communication time, i.e., sending values from one device to another as this highly depends on the setting in the field (e.g., power line communication, radio, ...) which is hardly reflected in the lab setup. For each smart meter in our lab setting we measure (i) the time t_M for computing $s_{i,t}$, $\tilde{m}_{i,t}$ and S_i; and (ii) the time t_H for computing $H(m_{i,t})$ and $H(s_{i,t})$. This measurement is performed for variable bit lengths $b \in \{32, 64, 128, 256, 512\}$. Table V shows the resulting values (median from 1000 samples). For larger bit sizes the homomorphic hash takes up most of the computation time, e.g., for $b = 256$ this yields the following measurements:

$t_M + t_H = 6.2048\text{ms}$ and the homomorphic hash has a fraction of $\frac{t_H}{t_M + t_H} = 94.95\%$.

TABLE V
COMPUTATION TIMES FOR THE HASH t_H AND FOR MASKING t_M FOR VARIABLE BIT LENGTHS b (MEDIAN, 1000 SAMPLES EACH).

Bit length b	t_H [ms]	t_M [ms]
32	0.1933	0.2435
64	0.3202	0.2437
128	1.0932	0.2597
256	5.8914	0.3134
512	37.4880	0.4674

In order to evaluate the behavior of the protocol when smart meters fail during one round of reading, 20, 40, 60, 80 and 100 smart meters are virtualized. For each group size, the time for one round of spatial aggregation is measured with all smart meters being available and with 10 smart meters randomly failing during normal operation. The methodology is the same as described for the lab setting above and the bit length is set to $b = 256$.

The timing results are shown in Figure 4. The box indicates the first and third quantile and the horizontal bar in the box is the median. The time needed for spatial aggregation increases with the group size. If 10 smart meters fail at random, the average time for aggregation is lower than if no smart meters fail. This is the case, since for 10 smart meters neither the hash nor the masking needs to be calculated. The time saved for calculating the masking of the value and especially the hash (compare Tables III and V) is greater than the additional time needed for skipping the smart meter and addressing the next one in the sending list. However, we do not consider network delays due to packet loss or low bandwidth (which may occur when using PLC or radio links in the field).

C. Network Traffic

In terms of network traffic, the message size is determined by the bit length b, that in turn provides the basis for the upper length of the values and the hashes. In Table VI an estimation is provided for the upper length of the messages, expressed as multiples of the bit length. The length of Ack messages and the message for requesting S_T are of negligible size. It can be seen, that initialization is achieved with low overhead and that temporal aggregation is very efficient in terms of network traffic. In addition, for one round of reading, the bulk of the communication is between smart meters and the data concentrator, whereas smart meters only exchange short messages.

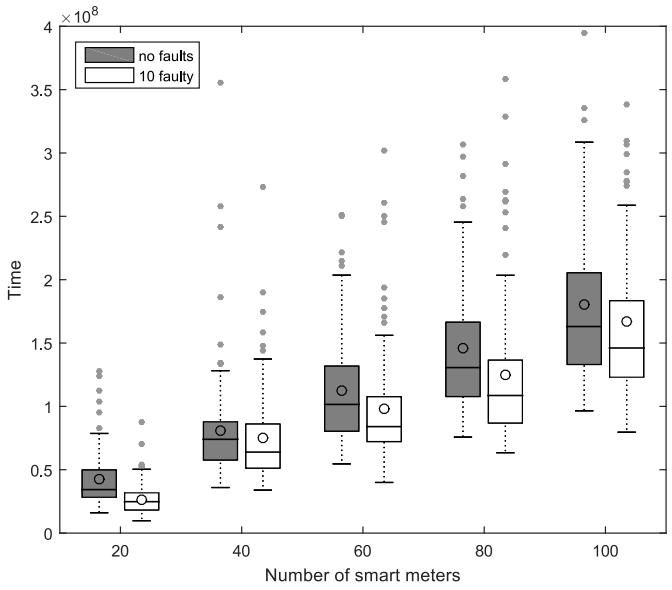


Fig. 4. Timing results for one round of reading with a group size of 20, 40, 60, 80 and 100 virtualized smart meters. Results are shown for no faults and for 10 randomly failing smart meters.The box indicates the first and third quantile and the horizontal bar in the box is the median.

TABLE VI
MESSAGE SIZES FOR ONE ROUND OF READING.

	Message	Bit length
$\text{DC} \rightarrow \text{SM}_1$	S_0	b
$\text{DC} \rightarrow \text{SM}_i$	Ack	
$\text{SM}_i \rightarrow \text{DC}$	$\tilde{m}_{i,t}\|\|H(m_{i,t})\|\|H(s_{i,t})\|\|t$	$4b$
$\text{SM}_N \rightarrow \text{DC}$	S_N	b
$\text{SM}_i \rightarrow \text{SM}_{i+1}$	S_i	b
$\text{SM}_{i+1} \rightarrow \text{SM}_i$	Ack	

VII. CONCLUSION AND FUTURE WORK

In this paper we discussed a novel approach for an error-resilient spatio-temporal masking protocol. The protocol is capable of privacy-aware aggregation over a number of smart meters in terms of space (as used for network monitoring) and time (as used for billing). While our basic algorithm still suffered from some flaws, such as only limited fault tolerance and the lack of an ability to proof correctness of shares, the final algorithm fulfills all of these requirements. Our protocol is designed for protecting end-user privacy and therefore we conducted a thorough privacy analysis with respect to honest-but-curious and passive adversaries. Evaluation has shown that this approach is feasible for practical implementations, especially as all computations on the smart meter are of constant time for normal operation and that the computation time for masking is negligible compared to the homomorphic hashes. The protocol which is presented in this paper contributes in terms of error-resilience and spatio-temporal masking. State of the art approaches have no or only limited support for efficiently treating with faulty smart meters.

For the proposed protocol, it will be interesting to increase security using, e.g., signatures that are specifically designed

for this situation. Additionally, verification could eventually be improved using, e.g., commitments.

ACKNOWLEDGMENTS

The financial support of the Josef Ressel Center by the Austrian Federal Ministry of Science, Research and Economy the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Funding by the Federal State of Salzburg is gratefully acknowledged.

REFERENCES

[1] G. Eibl and D. Engel, "Influence of Data Granularity on Nonintrusive Appliance Load Monitoring," in *Proceedings of the Second ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '14)*. Salzburg, Austria: ACM, 2014, pp. 147–151.

[2] ——, "Influence of Data Granularity on Smart Meter Privacy," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 930–939, 2015.

[3] S. Finster and I. Baumgart, "Privacy-Aware Smart Metering: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1732–1745, jan 2014.

[4] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 504–512.

[5] W. Yang, N. Li, Y. Qi, W. Quardaji, S. McLaughlin, and P. McDaniel, "Minimizing Private Data Disclosures in the Smart Grid," in *Proceedings of the ACM Conference on Computer and Communications Security 2012 (CCS'12)*. New York, NY, USA: ACM, 2012, pp. 415–427.

[6] L. Yang, X. Chen, J. Zhang, and H. V. Poor, "Cost-Effective and Privacy-Preserving Energy Management for Smart Meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 486–495, 2016.

[7] D. Chen, S. Kalra, D. Irwin, P. Shenoy, and J. Albrecht, "Preventing occupancy detection from smart meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2426–2434, 2015.

[8] C. Castelluccia, A. C. Chan, E. Mykletun, and G. Tsudik, "Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, 2009.

[9] Z. Erkin and G. Tsudik, "Private computation of spatial and temporal power consumption with smart meters," in *Proceedings of the 10th international conference on Applied Cryptography and Network Security*, ser. ACNS'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 561–577.

[10] F. Li, B. Luo, and P. Liu, "Secure Information Aggregation for Smart Grids Using Homomorphic Encryption," in *Proceedings of First IEEE International Conference on Smart Grid Communications*, Gaithersburg, Maryland, USA, 2010, pp. 327–332.

[11] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology — EUROCRYPT '99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, vol. 1592, pp. 223–238.

[12] G. Danezis, M. Kohlweiss, and A. Rial, *Differentially private billing with rebates*, T. Filler, T. Pevny, S. Craver, and A. Ker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6958 LNCS.

[13] G. Acs and C. Castelluccia, "I have a DREAM! (DiffeRentially privatE smArt Metering)," in *Proc. Information Hiding Conference*, 2011, pp. 118–132.

[14] E. Shi, R. Chow, T.-h. H. Chan, D. Song, and E. Rieffel, "Privacy-preserving aggregation of time-series data," in *Proc. NDSS Symposium 2011*, 2011.

[15] T. H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7397 LNCS, pp. 200–214, 2012.

[16] C. Efthymiou and G. Kalogridis, "Smart Grid Privacy via Anonymization of Smart Metering Data," in *Proceedings of First IEEE International Conference on Smart Grid Communications*, Gaithersburg, Maryland, USA, 2010, pp. 238–243.

[17] D. Engel and G. Eibl, "Multi-Resolution Load Curve Representation with Privacy-preserving Aggregation," in *Proceedings of IEEE Innovative Smart Grid Technologies (ISGT) 2013*. Copenhagen, Denmark: IEEE, 2013, pp. 1–5.

[18] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart grid," in *Privacy Enhanced Technology Symposium*, 2011, pp. 175–191.

[19] F. Gomez Marmol, C. Sorge, R. Petrlic, O. Ugus, D. Westhoff, and G. Martinez Perez, "Privacy-enhanced architecture for smart metering," *International Journal of Information Security*, vol. 12, no. 2, pp. 67–82, 2013.

[20] G. Danezis, C. Fournet, M. Kohlweiss, and S. Zanella-Béguelin, "Smart Meter Aggregation via Secret-sharing," in *Proceedings of the First ACM Workshop on Smart Energy Grid Security*, ser. SEGS '13. New York, NY, USA: ACM, 2013, pp. 75–80.

[21] M. Jawurek, M. Johns, and F. Kerschbaum, "Plug-in privacy for smart metering billing," in *Privacy Enhancing Technologies (PETS)*, 2011, pp. 192–210.

[22] E. McKenna, I. Richardson, and M. Thomson, "Smart meter data: Balancing consumer privacy concerns with legitimate applications," *Energy Policy*, vol. 41, pp. 807–814, 2012.

[23] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," *Proceedings - IEEE Symposium on Security and Privacy*, vol. 2004, no. Section VIII, pp. 226–239, 2004.

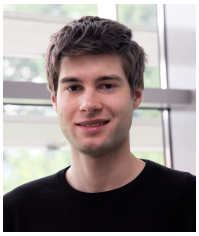
[24] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC Press, Inc., 1996, vol. 1.

[25] National Institute of Standards and Technology (NIST), "Specification for the Advanced Encryption Standard (AES)," 2001.

[26] ITU-T, "Recommendation ITU-T X.509 – Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks," 2012.

[27] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[28] M. LeMay, G. Gross, C. A. Gunter, and S. Garg, "Unified architecture for large-scale attested metering," *HICSS 2007. 40th Annual Hawaii International Conference on System Sciences, 2007*, pp. 1–10, 2007.



Fabian Knirsch (S'15) is a PhD student at the Computer Sciences Department at Salzburg University since 2015 and a researcher at the Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control at Salzburg University of Applied Sciences since 2014. He received the Master's degree *Dipl.-Ing.* in 2014 from Salzburg University of Applied Sciences. His current research interest is on methods and technologies that enhance security and privacy in the smart grid user domain.

Günther Eibl (M'13) received the Ph.D. degree in mathematics in 1997 and the M.Sc. degree in physics in 2002 from the University of Innsbruck, Innsbruck, Austria.

He is a research associate at the the Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control located at the Salzburg University of Applied Sciences in Austria. Previous research positions were located at the institutes of biostatistics, Innsbruck, Austria and the institute of theoretical physics, Innsbruck, Austria. In academic and non-academic research he worked in such fields as data mining and machine learning, particle and fluid simulations, computer vision, robot kinematics, control and cryptography. His research interests include extraction of information from data with a focus on statistical modeling, data mining and privacy preserving technologies.

Dominik Engel (S'06-M'08) received the Ph.D. degree in computer science from the University of Salzburg, Salzburg, Austria in 2008.

He is a Professor at the Salzburg University of Applied Sciences in Austria, where he heads the Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control. Prior to joining Salzburg University of Applied Sciences, Dominik Engel was a researcher at the Universities of Bremen, Germany and Salzburg, Austria and product manager at Sony DADC, Anif, Austria, where he was responsible for video content security. His current research interests include smart grid security and privacy and methods for enhancing trustworthiness of technical systems in general.