SMART-ER: peer-based privacy for smart metering

Sören Finster, Ingmar Baumgart
Institute of Telematics
Karlsruhe Institute of Technology (KIT), Germany
{finster,baumgart}@kit.edu

***Abstract*—Smart metering is an essential part of the future smart grid but causes privacy issues by collecting sensitive data from households with a high temporal resolution. Peer-based privacy mechanisms can solve this problem through privacy-aware aggregation. The SMART algorithm, originally proposed for wireless sensor networks, is a lightweight approach to this problem. In this paper, we propose to adapt the ideas of SMART for privacy-aware smart metering. However, our simulation results show, that accuracy suffers in the presence of communication errors especially in large networks. Therefore, we designed SMART-ER, an improved version of SMART. It utilizes dependency tracking and grouping to provide exact and robust smart metering even in the presence of communication errors. We show, that SMART-ER provides significantly more accurate results in typical churn scenarios.**

I. Introduction

An essential part of the transformation process from the classical power grid to a *smart grid* is the deployment of *smart meters* in all participating households. A smart meter periodically transmits the current power consumption of the corresponding household to, e.g. its energy provider. By this means, the energy provider gets an accurate view on the current power consumption of its customers.

A major drawback of smart metering from a customer perspective is a privacy leakage of sensitive personal data. This leakage includes, for example, working hours, vacations, habits and even religious beliefs. Therefore several mechanisms to protect customers' privacy have been proposed in the past. Most of these mechanisms depend either on a trusted third party or make use of computational intensive cryptography.

In this paper we propose a lightweight alternative without dependency on a trusted third party. Our peer-based privacy mechanism is based on the *Slice-Mix-AggRegaTe (SMART)* algorithm [1], originally published for data aggregation in wireless sensor networks. The main idea of SMART is to slice measurements into several pieces and to transfer (mix) each of these to different peers in the network. In a final step, slices get aggregated and are transmitted to the data sink.

Although this algorithm is very efficient and easy to implement, we show that it is error-prone in case slices get lost. Since smart meters are typically connected to the Internet using DSL or cable lines, packet loss or temporary loss of connectivity are errors which are expected to occur frequently.

Therefore we designed *SMART-ER (SMART for Exactness and Robustness)*, an enhanced version of SMART, which is robust against communication problems and temporary failures of smart meters. This is achieved by a changed slicing mechanism, by tracking of inter-node dependencies and by arranging nodes in smaller groups.

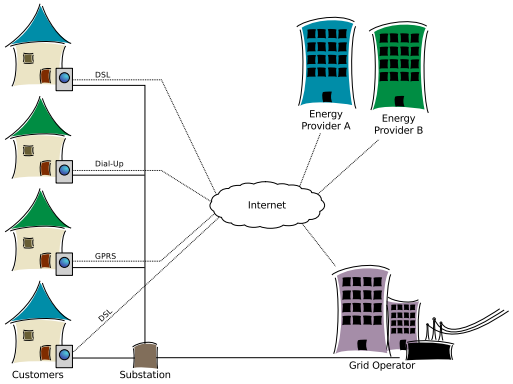

Fig. 1. Our scenario: energy providers have an interest in the power consumption of their respective customers for improved energy trading or energy production.

The rest of this paper is structured as follows: In Section II we provide an overview of related work. In Section III we describe our scenario and attack model. This is followed by a short introduction to SMART in Section IV including an analysis of shortcomings of SMART in the smart metering scenario. In Section V we describe the details of our SMART-ER algorithm followed by simulation results presented in Section VI. Finally, we conclude in section VII.

II. Related Work

Privacy-aware smart metering is often approached by using a trusted third party. In these approaches, the trusted third party is either used for pseudonymization (e.g. [2]) or privacy-aware aggregation of meter readings (e.g. [3]). Since a trusted third party induces operational costs and it is an open question why it should be trusted, there are several approaches to privacy-aware smart metering without a trusted third party (e.g. [4]–[7]). Yet, these proposals tend to rely on complex and computationally intensive cryptographic mechanisms (e.g. homomorphic encryption) and largely neglect problems that may arise through communication errors. In particular, there is no evaluation of the proposed approaches in the presence of communication errors.

To the best of our knowledge, SMART-ER is the first proposal that features a lightweight, privacy-aware scheme for smart metering without a trusted third party that is proven to provide robust aggregation results even in the presence of communication errors.

III. Scenario and Attack Model

In our scenario (see Figure 1), we assume that every customer can choose from a multitude of different energy

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

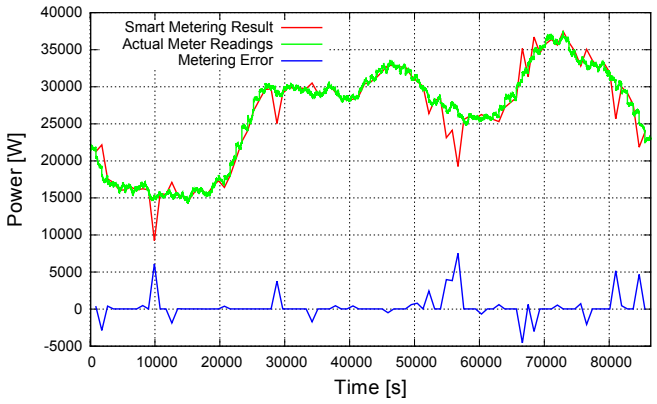

Fig. 2. Results of a day of smart metering of 50 households with SMART.

providers. Customers buy (and sell) their energy from (or to) an energy provider. The power grid itself is maintained by a power grid operator. Within the domain of a power grid operator, there can be a multitude of energy providers selling and buying energy to and from customers. Every household is equipped with a smart meter and every smart meter has some form of connection to the Internet. This enables communication with energy providers but also communication with other smart meters.

Energy providers have a strong interest in accurate and recent data about the power consumption of their respective customers for the purpose of trading at energy exchanges or providing real-time pricing. However, the customers do not want to give up their privacy for this. Since energy providers can achieve their goal with aggregated data, a privacy-aware, aggregating smart metering protocol is used.

Note, that this is solely for the smart metering application of monitoring. To use smart metering to bill customers, there are other means to provide a certain level of privacy. For example, by billing only monthly and aggregating the consumption on the smart meter.

The current consumption is measured by all smart meters with a fixed interval, for example every fifteen minutes. Measurements are roughly synchronized by the smart meters using a real-time clock. The time between measurements is called a *metering epoch*. The results of each epoch should arrive timely at the energy provider.

Different from other publications in privacy-aware smart metering, we do not propose the honest but curious attack model. Instead, we propose a stronger attacker. In our scenario, an attacker has the ability to eavesdrop on connections. However the complexity of eavesdropping increases exponentially with the number of connections. Additionally, we include the ability to insert malicious participants. Again, the complexity of this attack increases exponentially with the number of malicious participants. Finally, we also include the data sink as a potential attacker.

For the remainder of this paper, we will call smart meters "*nodes*" and the energy provider the "*data sink*".

IV. SMART

The SMART algorithm as described by He et al. [1] was originally designed for privacy-aware aggregation in wireless sensor networks. It is a peer-based approach in which privacy-sensitive information is spread randomly among a set of peers. Every peer aggregates incoming data and finally submits this aggregate to a data sink. We provide a short description of the algorithm in this section. For a more thorough description we refer to [1].

Let Φ be the set of all nodes in the network. Every node $v \in \Phi$ has a privacy-sensitive secret p_v. The data sink is interested in the sum $P = \sum_{v \in \Phi} p_v$.

The SMART algorithm is divided in three phases:

a) Slicing: In the first step, every node v randomly selects N other nodes from Φ with N being a configurable parameter of the algorithm. We call this set of nodes S_v. Then, v slices its private data randomly into $N+1$ slices s_{vw} with $w \in S_v \cup \{v\}$, such that $p_v = \sum s_{vw}$. Node v now has $N+1$ slices, one for each node in S_v and one for itself. Each slice s_{vw} is then sent to the corresponding node w. The slice s_{vv} is kept at the node itself. With $s_{vw} = 0$ if v did not send a slice to w, the desired sum at the data sink can now be expressed as $P = \sum_{v,w \in \Phi} s_{vw}$.

b) Mixing: Every node v waits for slices from other nodes for a certain time and aggregates the received slices to $r_v = \sum s_{wv}$ for all $w \in \Phi$ with $s_{wv} = 0$ if $v \notin S_w$. After that, v submits the sum $R_v = r_v + s_{vv}$ to the data sink.

c) Aggregation: The data sink receives the values R_v for each node $v \in \Phi$ and sums them up. It follows, that $\sum_{v \in \Phi} R_v = \sum_{v,w \in \Phi} s_{vw} = \sum_{v \in \Phi} p_v = P$ and therefore the sum in which the data sink was interested.

Due to the irreversibility of summation, the data sink is not able to extract information about single nodes from the aggregate.

A. Problems of SMART for energy related applications

For smart grid applications using smart metering results, exactness and robustness are very important features. Decisions with major implications about energy production, trading or pricing are influenced by the results of smart metering. The SMART algorithm achieves privacy with small computational overhead. But, as we show in Section VI, it introduces inexactness in the presence of communication errors. If a node was able to distribute slices but is not able to send the final submission, the aggregated sum will be randomly distorted.

Consider Figure 2 which depicts a day of smart metering in a small electrical grid[1]. The plot shows the results of smart metering, the actual measured values and the difference between those values as error. A positive error means, that smart metering signaled a power consumption lower than the real one, a negative error a higher one. As can be seen,

[1]We will use sample plots to illustrate features during the next sections. Used parameters are: 50 households, 3 000 maximum slice offset, $N = 2$, 24h average lifetime and 5min average recovery time

even in a small grid, errors occur frequently. Note, that errors are both, positive and negative. The data sink can determine how many submissions from nodes are missing simply by counting the number of final submissions. But since this means, that the aggregated result is randomly distorted, there is no deterministic way of recovering from this situation.

V. SMART-ER

The SMART-ER algorithm (SMART for Exactness and Robustness) is an improved variant of SMART for efficient privacy-aware smart metering. It is robust against node failures and communication problems. Through removal of distorted submissions, SMART-ER achieves exact results for the remaining submissions.

The main changes from SMART to SMART-ER are:

- A changed slicing mechanism using completely randomized data
- Tracking of inter-node dependencies and removal of unmet dependencies from the aggregate
- Arrangement of nodes in smaller groups to contain error spreading
- Extrapolation to deal with lost submissions

We describe each of those changes in the following subsections and will use sample plots from the following evaluation to illustrate the introduced changes.

A. Completely randomized slicing

As described in section IV, the SMART algorithm assumes a certain order of events. At first, each node chooses N targets for its slices (s_{vw}), then slices its private data in $N+1$ slices and finally sends one slice to every target in S_v, keeping one slice for itself. Since slices should not contain information about private data, it is important to slice randomly. The slicing technique proposed in [1] uses p_v for calculation and leaks information about it in its slices ($\forall s_{vw} : |s_{vw}| \leq p_v$).

In SMART-ER, there is no assumption of knowing the complete set S_v or the value p_v in advance. On the contrary, it is encouraged to build this set dynamically over time and send slices immediately to a target. Instead of slicing p_v to generate slices, we send out completely randomized slices to other nodes. Therefore, sent out slices are completely independent from p_v and, in fact, carry nothing but randomness.

The calculation of slices in SMART-ER is done as follows: For every slice that gets sent out from node v to another node w, a random positive or negative number s_{vw} is drawn. Node v maintains an aggregation of all sent out slices $RS_v = \sum_{w \in \Phi} s_{vw}$ and additionally all received slices r_v as before. When submitting the final value, each node v calculates the value to submit as $R_v = r_v + p_v - RS_v$. Effectively, every slice is subtracted by the sending node and added by the receiving node. Both operations cancel each other out in the final aggregate and leave only the aggregated private values.

Note, that these changes in the algorithm do not change the privacy preserving features. The analysis for privacy preservation in [1] still holds for this variant.

For real world application, it is necessary to keep arithmetic overflow in mind. Especially, with the complete randomization of slices, the range from which the random numbers are drawn is important. To prevent arithmetic overflow, the random numbers must be significantly smaller than the registers for RS_v, R_v and the final submission. In our implementation, we use a 32 bit representation for slices and a 64 bit representation for all aggregating registers. Therefore, we draw random numbers for slices at maximum from the range $[-2^{31}, 2^{31}-1]$. We will call the positive number of this range the *slice offset*.

With completely randomized slicing, we decoupled slicing from private data and, in fact, can perform slicing without knowing the private data. This enables a feature we call *pre-slicing*: Instead of waiting for the metering epoch to start and then perform SMART-ER, we perform most of the algorithm before the actual metering epoch started. Slices are sent out, received and aggregated before the actual private data is generated. If all slices are sent out and received at the moment of measurement, the final value can be calculated immediately and submitted to the data sink. Pre-slicing reduces the delay at the data sink by the time, the slicing and mixing process takes. It provides the data sink with valuable information at the earliest possible point in time. Additionally, pre-slicing enables to do slicing in advance for several metering epochs. For example, nodes could perform pre-slicing for all metering epochs of a coming day at midnight.

B. Dependency Tracking

To remove random distortion from the final aggregation, SMART-ER uses dependency tracking. For each slice that is sent out or received by a node, the corresponding node is added to a dependency list. When a node sends the final submission to the data sink, it includes its dependency list.

The data sink is now able to calculate an exact result. For this, we define $\rightsquigarrow$ as the relation "sliced to". If node $v, w \in \Phi$ and $v \rightsquigarrow w$ then node v sent a slice to node w during the slice-mix phase. This implies a dependency from node v to node w concerning the correctness of v's submission and as well a dependency from node w to node v concerning the correctness of w's submission. If w does not submit a value, the submission of v will be incorrect since it lacks the slice sent to w. On the other hand, if w submits a value but v does not, w's value contains a slice without counterpart and will be incorrect as well. Further, if w submits a value but a node $x \in \Phi$ with $w \rightsquigarrow x$ does not, not only w's value will be incorrect but also v's value will thereby be incorrect. Following this argument, we define the relation "depends on" ($\leftrightsquigarrow$) as the symmetric, reflexive, transitive closure of the relation $\rightsquigarrow$ on Φ. So, $\leftrightsquigarrow$ is the equivalence relation generated by $\rightsquigarrow$.

Most of the nodes submit their value to the data sink. But due to communication problems some may be missing. Therefore, we define $S \subseteq \Phi$ as the set of nodes that submitted a value to the data sink. To guarantee accurate results, we need to find S', the largest subset of S, whose nodes do not depend on nodes outside of S'. We claim that $S' = \{v \in S \mid \forall w \in \Phi : v \leftrightsquigarrow w \Rightarrow w \in S\}$ fulfills this requirements.

We have to prove, that for all $v \in S'$ and $w \in \Phi$ holds $v \leftrightsquigarrow w \Rightarrow w \in S$. Suppose that there exists a $v \in S'$ and a $w \in \Phi \setminus S'$ with $v \leftrightsquigarrow w$. By construction of S', this implies

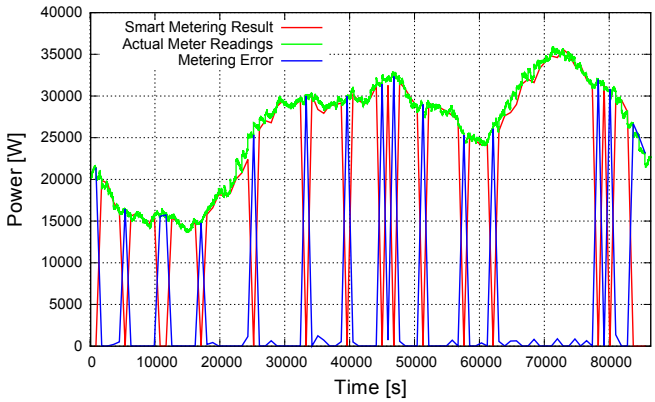

Fig. 3. Results of a day of smart metering of 50 households with dependency tracking.

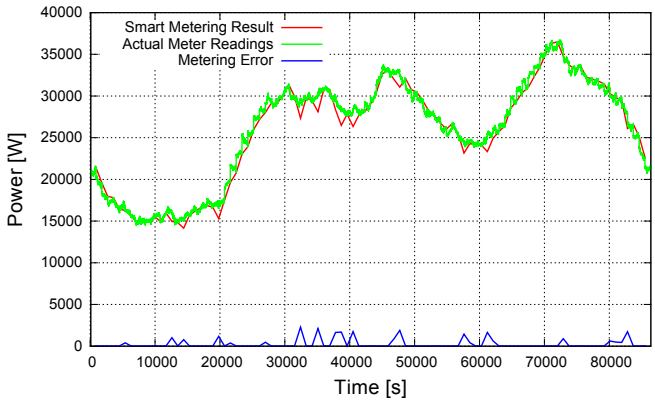

Fig. 4. Results of a day of smart metering of 50 households with dependency tracking and grouping.

that $w \in S$. As w is not contained in S' there exists $x \in \Phi \setminus S$ with $w \leftrightsquigarrow x$. So we have $v \leftrightsquigarrow w$ and $w \leftrightsquigarrow x$. By the transitivity of $\leftrightsquigarrow$, it follows that $v \leftrightsquigarrow x$. As $x \notin S$ this implies $v \notin S'$ which contradicts our assumption.

It remains to show that S' is the largest subset that fulfills this requirement. This follows immediately because every $v \in S \setminus S'$ depends on a node which is not even in S. □

Therefore, after receiving all submissions, the data sink calculates the symmetric, reflexive, transitive closure of the submitted dependencies and removes all values whose dependencies could not be fulfilled.

The effect of dependency tracking is shown by simulations results in Figure 3. Note, that there is no negative error since all invalid submissions are removed. Small positive errors depict situations, in which a small number of submissions had to be removed since their dependencies could not be fulfilled. But, as can be seen in the plot, in many cases the transitive closure of dependencies spanned all submissions resulting in a complete loss of information. Therefore, the error is as high as the actual metering result and the metering result is zero.

With dependency tracking activated, we have the advantage of ruling out random distortion. A result is always exact for the remaining submission. But the removal of all submissions is an unwanted effect we will address in the next section.

C. Grouping

As we saw in Figure 3, the set S' can get very small or be even empty very easily. The cause for this lies in the random distribution of slices within all nodes of the network. Through random distribution, the equivalence class of a node v, $[v] = \{w \in \Phi \mid v \leftrightsquigarrow w\}$ can get very large. If one of the nodes in $[v]$ fails to submit a value, $[v]$ is excluded from S'.

To reduce the maximum size of equivalence classes, we introduce grouping. The grouping feature of SMART-ER creates disjoint sets $G_i \subseteq \Phi$ called "groups". Each node v of such a group exchanges slices only with other members of the same group. It follows, that $\forall v \in G_i : [v] \subseteq G_i$ meaning that dependencies do not cross group borders.

For evaluation, we implemented a simple version of grouping where the data sink assigns each node a group. The group size is chosen to be one more than the configured number of slices to send out. For example, if SMART-ER is configured to send out 2 slices, the data sink will build groups of size 3. Unless a node failure happens, this results in $\forall v \in G_i : [v] = G_i$.

In Figure 4, we see a sample plot with grouping and dependency tracking enabled. Missing submissions now are contained to smaller groups and therefore result in small, positive errors.

Note, that from a privacy perspective, group organization by the data sink is not optimal and will be further discussed in Section VI-C.

D. Extrapolation

As we saw in Figure 4, we now have exact information about a large subset of Φ. Under the assumption, that this large subset is representative for the missing nodes, we can now extrapolate the results to the total number of nodes. Since the data sink knows the expected number of nodes, it can calculate an extrapolated aggregation.

In our implementation, this is done only if we have exact information about at least half of the expected nodes. The resulting load profile is depicted in Figure 5. As can be seen, the error is very small compared to initial values. Note, that extrapolation allows negative errors again since the extrapolated data may not represent the missing nodes perfectly.

VI. Evaluation

We implemented SMART and SMART-ER – both using the improved slicing technique – in the overlay network simulator OverSim [8], which we extended with a model for power grids. We used OverSim's SimpleUnderlay as communication network, simulating typical Internet behavior with DSL connections. Households are simulated using standard load profiles from BDEW, the German Association of Energy and Water Industries. Our churn simulation is according to German broadband characteristics [9] under the assumption, that users do not manually disconnect. If not stated otherwise, we used an average connection lifetime of 24 hours and a recovery time of 5 minutes with variations using OverSim's LifetimeChurn. We configured both, SMART-ER and SMART to send out two

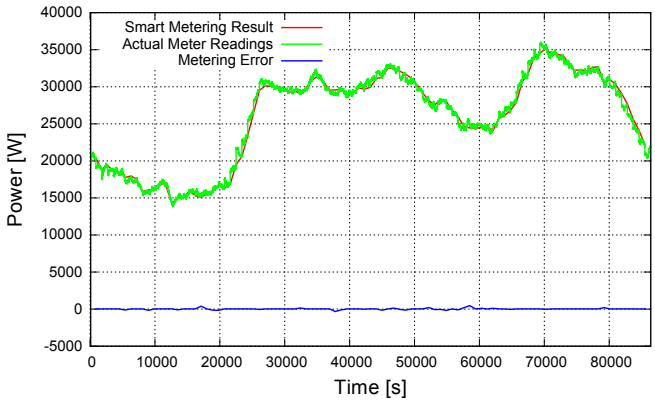

Fig. 5. Results of a day of smart metering of 50 households with SMART-ER (dependency tracking, grouping and extrapolation).

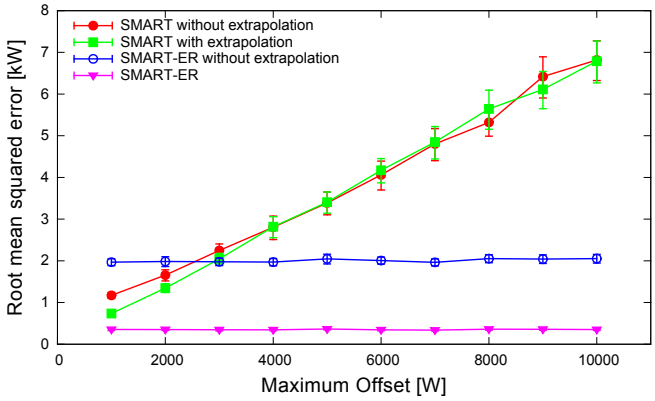

Fig. 6. Smart metering error depending on slice offset. Simulated for 100 nodes, 2 slices and an average node lifetime of 24h.

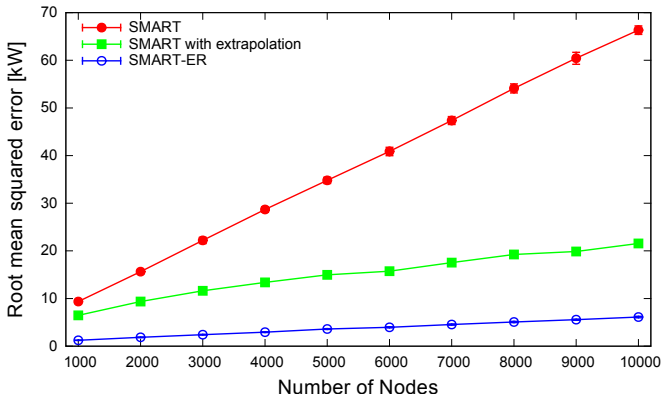

Fig. 7. Smart metering error depending on number of nodes. Simulated for offset 3000, 2 slices and an average node lifetime of 24h.

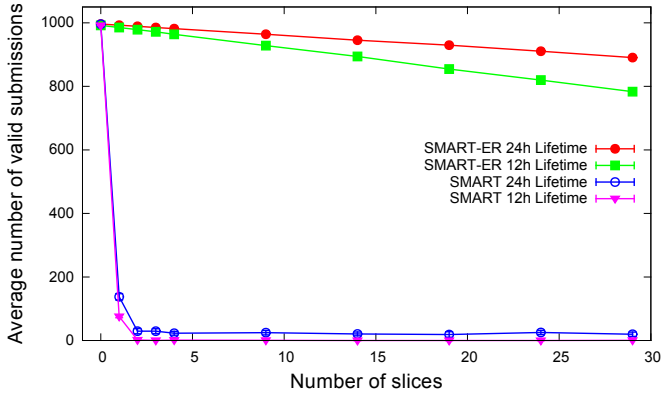

Fig. 8. Valid Submissions depending on number of slices. Simulated for 1000 nodes and average node lifetimes of 24h and 12h.

slices with configurable offset to other nodes, according to the recommendation in [1] and the mixing process had a duration of 5 minutes. Each run was repeated 30 times and we plotted the mean values with 98% confidence intervals.

A. *Precision of smart metering*

If we look at smart metering as a way of estimating the real power consumption, we can use an evaluation metric for statistical estimators, the root-mean-square error (RMSE). The error of a smart metering result is in our scenario the difference between the actual power consumption at the time of measurement and the reported power consumption. Without churn, the error would be zero. Note, that latency of smart metering results has no influence on the error.

First, we look at the completely randomized slicing feature and see in Figure 6, that SMART-ER errors do not depend on the maximum slice offset. The error of SMART-ER without extrapolation represents the missing smart meters due to churn. In the same figure, we see that SMART's errors increase with larger slice offsets since it cannot recover from missing slices. Also, extrapolation for SMART provides in general slightly improved results but has a great variance with higher offsets. For the following evaluations, we used a maximum offset of 3000W for both, SMART and SMART-ER, as a compromise between privacy and accurate SMART results.

In Figure 7, we evaluated the scalability of SMART-ER in the number of smart meters. For SMART, a higher number of participating smart meters results in a larger error since more slices get lost on average. Extrapolating improves the results. SMART-ER also suffers from the larger number of missing slices but recovers significantly and consistently better.

Figure 8 shows the reason for that. We simulated several configurations for the number of sent out slices for two churn scenarios, 24h lifetime and 12h lifetime. We observed how many submissions were valid by removing the submissions with unmet dependencies. As can be seen, with SMART only around 10% of submissions are exact. The other 90% did not satisfy their dependencies. A higher churn rate makes this even worse. SMART-ER is also sensitive to larger numbers of sent out slices since this effectively increases $[v]$. Yet, even with very large numbers of sent out slices and high churn, SMART-ER manages to achieve around 80% of exact results.

Finally, we took a closer look at higher churn rates. Figure 9 shows our results for churn rates as low as 6 hours average lifetime. As can be seen, SMART suffers under higher churn rates since more slices are missing on average. SMART-ER is able to recover from those losses and has a comparatively small error even in high churn scenarios.

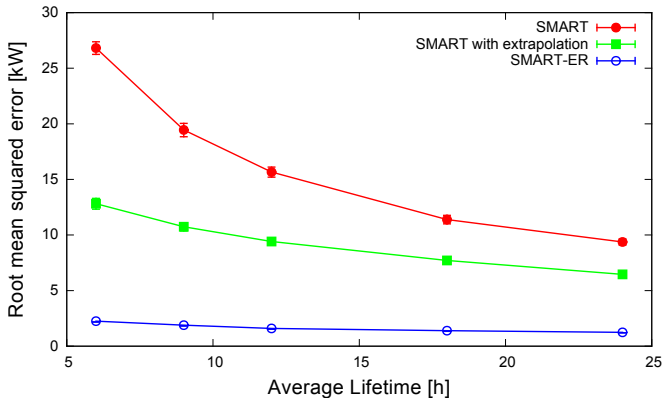

Fig. 9. Smart metering error depending on churn. Simulated for 1000 nodes, offset 3000 and 2 slices.

B. Communication Overhead

Due to dependency tracking SMART-ER has a higher communication overhead than SMART. In the message with the final submission to the data sink, all dependencies of that submission have to be listed. This includes all dependencies for sent out slices and all dependencies for received slices and therefore is highly dependent on the configured number of slices a node sends out. With every dependency, one identifier for the dependency has to be added to the message. If, for example, the IPv6 address is used as identifying information, this adds 16 bytes per dependency to the submission message.

However, the added dependencies is limited through the grouping feature to one less than the size of the group. For example, if four slices are configured, each group has a size of five. Therefore, the maximum number of dependencies a node can have is four other nodes or $4 \times 16 = 64$ Bytes. Even for severely constrained connections, this overhead is negligible.

C. Achieved privacy

In this section, we want to take a short look at the privacy achieved by SMART-ER.

Most of SMART-ER's features do not interfere with the privacy guarantees from SMART. In fact, completely randomized slicing and pre-slicing even improve those privacy guarantees by making sure, that the slices themselves do not transport sensitive information.

Dependency tracking provides the data sink with more information than SMART. The dependency list tells the data sink who peered with whom. By itself, this is not a problem for privacy. But if this is a concern, it can be avoided by using short-lived, randomly generated identifiers for dependency lists and some sort of sender obfuscation to prevent the data sink from learning the transport address of the source, e.g. onion routing.

However, the grouping feature opens up an attack vector for the data sink through the organization of groups. If the data sink has total control over groups, it could assign a targeted smart meter a group that consists only of malicious smart meters that cooperate with the data sink. This would allow the data sink to break the privacy of a targeted smart meter. But this also means, that the data sink has to have enough malicious smart meters at disposition to fill one group. This may be a difficult attack especially if SMART-ER's ability to work well with larger group sizes is taken into account. Also, there are possibilities to arrange these groups without total control from the data sink, e.g. [10], [11].

VII. Conclusion

Smart metering is a central component of future smart grids and enables the energy provider to closely monitor power consumption in every part of the grid. However currently deployed smart metering solutions threaten the privacy of the customers. The SMART algorithm provides a lightweight and efficient approach to privacy-aware data aggregation in wireless sensor networks. In this paper we showed that SMART can also be utilized for privacy-aware smart metering without a trusted third party. However we showed that SMART produces unreliable results in case of communication problems or smart meter failures. Therefore we propose an enhanced algorithm called SMART-ER. SMART-ER provides the same level of privacy as SMART but achieves much more accurate results in case of communication errors or node failures. Our simulation results show that in typical churn scenarios SMART-ER achieves consistently better results than SMART with negligible communication overhead.

References

[1] W. He, X. Liu, H. V. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-Preserving Data Aggregation for Information Collection," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 1–22, Aug. 2011.

[2] C. Efthymiou and G. Kalogridis, "Smart Grid Privacy via Anonymization of Smart Metering Data," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, Oct. 2010, pp. 238–243.

[3] J.-M. Bohli, C. Sorge, and O. Ugus, "A Privacy Model for Smart Metering," in *IEEE International Conference on Communications Workshops (ICC)*, Capetown, May 2010, pp. 1–5.

[4] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *International Journal of Security and Networks*, vol. 6, no. 1, p. 28, 2011.

[5] F. Mármol, C. Sorge, O. Ugus, and G. Pérez, "Do not snoop my habits: preserving privacy in the smart grid," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 166–172, May 2012.

[6] F. D. Garcia and B. Jacobs, "Privacy-Friendly Energy-Metering via Homomorphic Encryption," in *Proceedings of the 6th international conference on Security and trust management*, Surat, May 2011, pp. 226–238.

[7] F. Gómez Mármol, C. Sorge, R. Petrlic, O. Ugus, D. Westhoff, and G. Martínez Pérez, "Privacy-enhanced architecture for smart metering," *International Journal of Information Security*, vol. 12, no. 2, pp. 67–82, Nov. 2012.

[8] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," in *2007 IEEE Global Internet Symposium*. IEEE, May 2007, pp. 79–84.

[9] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic," in *Proceedings of the 9th ACM Internet measurement conference (IMC '09)*. Chicago, Illinois: ACM, Nov. 2009, pp. 90–102.

[10] S. Finster and I. Baumgart, "Elderberry: A peer-to-peer, privacy-aware smart metering protocol," in *IEEE INFOCOM Workshop on Communications and Control for Smart Energy Systems*, Turin, Apr. 2013, pp. 3411–3416.

[11] S. Finster, "Smart Meter Speed Dating, short-term relationships for improved privacy in Smart Metering," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Vancouver, Oct. 2013, pp. 426–431.