Anonymous Data Collection for the Smart Grid

Edward Tremel, Ken Birman, and Bobby Kleinberg
Cornell University, Ithaca, NY 14850
Email: {edward,ken,rdk}@cs.cornell.edu

Márk Jelasity
Hungarian Academy of Sciences
and University of Szeged, Hungary
Email: jelasity@inf.u-szeged.edu

***Abstract*—The smart grid has great potential to improve the reliability and efficiency of power distribution by providing timely, accurate information to utility companies and allowing for cooperative control of demand at the client side. However, naive implementations of smart grid data collection could jeopardize the privacy of consumers, and concerns about privacy are a significant obstacle to the rollout of smart grid technology. We propose a communication system for smart meters that allows the utility owner to query meter data and publish demand-response commands without compromising the privacy of participating customers. Our protocol is scalable to networks with hundreds of thousands of nodes, and requires only basic cryptography primitives that can feasibly be implemented on the limited hardware available to smart meters.**

I. Introduction

The smart grid is expected to provide many exciting opportunities for improving the efficiency, reliability, and sustainability of electrical power transmission. Unlike traditional analog meters, smart meters can be used to continuously measure, predict, and even control power consumption within individual homes and businesses, and a grid containing smart meters can use this data to more dynamically and accurately adapt power generation to power use. This can help utilities avoid unnecessary power generation and unexpected overloads, and allow consumers to reschedule their power consumption to save money. Recent US government studies have concluded that demand-response systems that reduce variation in load using the advanced metering infrastructure [1] have the potential to reduce peak load by up to 20% during summer months [2]. The advanced features of smart meters all depend on active communication between the utility company and the meters many times during the day. Unfortunately, customer concerns about the privacy of their electrical power data have lead to protests, boycotts, and in some cases outright bans on smart meters that have two-way communication capabilities [3].

Consumers are right to be worried about their privacy, though, because the fine-grained power usage data collected by smart meters can leak a surprising amount of personal information. Experiments in Non-Intrusive Load Monitoring (NILM) [4] show that the time of use of individual electrical appliances can be extracted from meter data, and this information can be used to infer much about the personal habits of the home's occupants. Worse, the current approach of most smart grid projects is to send this fine-grained smart meter data directly to a centralized database at the utility, where it can easily be accessed by employees and government regulators [5].

The power systems research community has proposed several possible designs for a smart metering system that protects the privacy of consumers, using tools such as homomorphic encryption [6], secret sharing [7], value masking [8], and differentially-private noise [9]. However, these solutions tend to either assume that smart meters are powerful enough to perform advanced cryptography, or do not allow queries more complex than a simple addition of all contributed meter values. In this paper, we describe a different approach to privacy-preserving smart metering, based on onion routing over a deterministic peer-to-peer network. Our system relies on more practical assumptions about the computational abilities of smart meters, allows for more expressive queries, and still produces accurate results in the presence of failures. It will allow smart grid operators to get accurate, up-to-date information about the state of the grid, including current power usage and available power rescheduling capacity, without compromising the privacy of consumers.

II. Background and Related Work

The research community has been aware of the privacy problems related to smart metering for several years now. Research in NILM has shown, in work such as [4], that the time of use of individual electrical appliances can be extracted from meter data, and this information can be used to infer much about the personal habits of the home's occupants. Anderson and Fuloria pointed out in [5] that the current approach of most smart grid projects is to send all fine-grained smart meter data directly to a centralized database at the utility, where it can easily be accessed by employees and government regulators. Worse, in [10], Rouf et al. found that a widely-used implementation of the AMI standard for wirelessly transmitting meter data lacks basic security measures, so most meters upload their data unencrypted over insecure channels. McDaniel and McLaughlin [11] surveyed many of the security and privacy concerns that can arise in smart grids, and Lisovich and Wicker [12] identify many of the concrete technical challenges to preserving privacy.

Recent proposals for privacy-preserving smart metering are summarized by Erkin et al. in [13]. These generally fall into four major categories: homomorphic encryption schemes, secret sharing, value masking, and differential privacy.

Homomorphic encryption refers to cryptosystems in which mathematical operations can be executed on encrypted data, producing a ciphertext that decrypts to the results of applying

the same operations to the unencrypted data. Proposed smart-metering systems that use homomorphic encryption, such as [6] and [14], rely on partially homomorphic encryption (PHE) that allows only addition of encrypted data, because fully homomorphic encryption (which would allow any function to be executed on encrypted data) is still exponentially slow. In these systems, each meter encrypts its data under the homomorphic encryption, the encrypted energy consumption data is summed together by either the utility (as in [6]) or intermediate meters in an aggregation tree (as in [14]), and only the final sum is decrypted by the utility. Unfortunately, even though PHE is more efficient than fully homomorphic encryption, the encryption functions are still computationally expensive, and computing addition on a ciphertext is still slower than addition over the equivalent plaintext, partially because ciphertexts are orders of magnitude larger than plaintexts.

Secret-sharing, in which a meter's value is split into multiple shares that can only re-create the value when combined in the aggregate, can also be used to preserve privacy. Rottondi et al. [15], in addition to describing a PHE-based system similar to He et al.'s, propose a system in which each smart meter sends a share of its data to several "gateway" nodes (which are intermediate between the meters and the utility). The gateway nodes sum the shares and send the results to the utility, which combines the summed shares to recover the correct sum. Garcia and Jacobs [7] combine a secret sharing scheme with partially homomorphic encryption to produce what is essentially a special-purpose secure multiparty computation.

Another alternative to homomorphic encryption is value masking, such as the system proposed by Kursawe et al. in [8]. In value masking systems, each meter adds a secret, random "mask" value to the value it contributes, chosen such that the sum of all mask values is zero. The difficulty comes in generating the masking values in a way that ensures they will sum to zero, which requires coordination between the meters, without revealing which meter will use which masking value. Kursawe et al. present several options for doing this, including a secret-sharing scheme and an adaptation of the Dining Cryptographers protocol.

PHE, secret-sharing, and value masking all suffer from the limitation that they can only be used to compute one type of query over the meters' inputs, namely the sum function.Another approach that potentially allows for more nuanced queries is to apply techniques from Differential Privacy, a framework first proposed by Dwork in [16]. In differentially private systems, a small amount of noise is added to the result of each query run on a database, to ensure that a curious adversary cannot analyze query results to determine any particular individual's contribution to the database. The amount of noise added is proportional to the sensitivity of the query (its ability to reveal the effect of one individual on the result), so any query can be used as long as its sensitivity can be quantified. However, Differential Privacy assumes that the database is stored on a trusted party, which adds the noise before revealing the query results. In order to use Differential Privacy for smart metering, a trusted third party would need to add the noise, because it must be added after the query result is computed but before the utility provider sees it. Ács and Castelluccia [9] propose a system in which noise is added to each meter's measurement locally before it is contributed to the aggregate, but they also use additive PHE to keep these noisy values hidden from the utility until the aggregate is computed, because individual noisy values do not preserve privacy. As a result, their system is still limited to sum queries.

General-purpose secure multiparty computation, such as the garbled-circuits approach presented in [17], would allow for any query to be executed over the encrypted meter data. However, even the most efficient MPC algorithms require several expensive operations on ciphertexts per gate of the circuit representing the query, and the size of the circuit grows with the number of contributors to the computation [18]. Thus, this approach would not scale to the large numbers of participants required for smart metering, and would be infeasibly slow to implement on smart meters.

We propose a new approach to privacy-preserving smart metering that differs from all of these. Our goal is to provide a reasonable level of privacy to consumers without requiring smart meters to do expensive cryptography or restricting the system to only a single type of query over the data.

III. Our Approach

In this section we will briefly describe our proposed scheme. For reasons of brevity, we omit the details of its implementation, which we describe at length in [19].

A. System Model

Our design targets a smart grid deployment in which all meters are connected to a network and can communicate both with the utility owner and with each other. However, it is not necessary for the correctness and security of our protocol that the meters be able to communicate directly with each other; meters may connect to each other by routing through a central hub at the utility or through a network switch at the substation level. In order to verify the authenticity of messages and allow secure communication channels to be established, we assume that each smart meter has a private key it can use for signing messages, and that the utility owner keeps a database of all the meters' corresponding public keys. Other secure smart metering systems, such as [8] and [7], have made similar assumptions, and we believe it is reasonable to expect that utility owners will want to give their meters private keys in order to ensure the integrity of meter readings.

We do not assume that smart meters have significant computational capabilities, except for the basic public-key (RSA) and symmetric (AES) encryption capabilities needed to support standard network security (i.e. TLS). Since these encryption functions are industry standards used in a wide variety of applications, they may be implemented with dedicated hardware or specialized software designed for low-resource systems.

B. Threat Model

As with most other smart metering systems [13], we assume that the utility operator is honest-but-curious (or semihonest),

and so will attempt to learn everything he can about consumers' private data without disrupting the correct functioning of the system. We also assume that, from the perspective of any one customer, the other smart meters in the system are honest-but-curious, since consumers who are uncomfortable with the utility learning about their private data would be equally uncomfortable with their neighbors learning it. Our protocol can be modified to tolerate a small number of actively malicious smart meters (i.e. Byzantine faults), but that increases its complexity and running time, so for reasons of brevity we omit the Byzantine-fault-tolerant version here.

We do not assume, however, that smart meters are completely reliable; up to t of them may crash during a query. A crash failure includes any situation in which the meter stops sending and receiving messages, whether due to loss of power, interruptions in network connectivity, or software failure on the meter. Our protocol can be adjusted to prioritize either communications efficiency or fault-tolerance, depending on the expected value of t. One option, which we will explain carefully, is ideal if the number of failures is expected to be $O(\log n)$. This scheme is suitable even in low-bandwidth settings because each meter needs to transmit only a small fraction of the values being collected. However, it becomes very slow if t is a substantial percentage of n. For high failure rates, we offer a second, simpler option based on flooding values through the network.

C. Protocol Design

To set up our system, the utility assigns each smart meter a unique integer ID between 0 and $n-1$, where n is the number of smart meters, and publishes a data table mapping each integer ID to a meter's public key certificate. Meters may download this table locally, or query it from the utility when they need to learn which meter corresponds to which ID, but since the number of meters in the system should change only rarely we consider this information essentially static.

Once the meters have been assigned IDs, the utility can begin issuing queries. Smart meters respond to a query in three phases: shuffle, echo, and aggregate.

Shuffle: Upon receiving a query from the utility, each meter picks $t+1$ proxy meters, choosing one at random from each sequence of $n/(t+1)$ consecutive meter IDs. It then sends the value it will contribute to the query, plus the list of chosen proxies, to each proxy by routing it through at least 3 other meters in a peer-to-peer overlay. The peer-to-peer overlay is constructed as follows: In round j of message exchange, the meter with ID i sends a message to the meter with ID $i + 2^j \mod n$. (The round number starts at 0 at the beginning of each query, and is tracked asynchronously and independently by each node). The sending meter picks $t+1$ independent paths to its proxies through this overlay, where a "path" is a sequence of transitions through the overlay, either remaining at the same node for a round or continuing along its single forwarding path for that round. Before sending the values, the sending meter encrypts them with an "onion" of public-key encryptions, where each layer corresponds to one meter along the path and can only be decrypted by the private key of that meter (similar to onion routing [20]). Each layer of encryption above the last contains only the ID of the next node the message should be forwarded to.

Echo: To complete the Shuffle phase, the meters run the peer-to-peer overlay for $t \log n$ rounds of communication, at which point the properties of our peer-to-peer overlay guarantee that t messages from each meter could have reached any other meter along independent paths.[1] However, not all proxy values may have reached their destination due to the t meter failures. Thus, in this phase each meter forwards a copy of each proxy value it holds (most meters will be a proxy for more than one value) to the t other meters that should proxy the same value, also using the peer-to-peer overlay. Onion encryption is not necessary, since the values are now being sent from a proxy, not the meter that contributed them. However, meters should still use secure and authenticated channels (e.g. TLS) to send messages in the overlay, especially if they are communicating over a network that is owned by the utility.

Aggregate: After another $t \log n$ rounds of peer-to-peer message forwarding, all echoed values will have reached their destinations. At this point, each group of $n/(t+1)$ meters conducts a simple binary-tree aggregation protocol to collect all the values and answer the query. As long as the query function is commutative and associative, each node in the tree can apply the function to its local values and send the partial result to its parent. The root of the aggregation tree in each group sends the final result back to the utility, along with a count of how many values contributed to that result. The utility should accept the result that has the most contributions.

Brevity precludes an equally detailed discussion of our solution for high failure rates that could involve a significant percentage of the total set of meters. In quick summary, for this method we have source meters select proxies as above, but now we use the overlay to carry out a broadcast rather than finding independent paths. Our overlay is a highly connected graph and hence even with many failures (e.g. 10-20%), a broadcast by flooding will reach all nodes in $O(\log n)$ hops with very high probability. The messages are still encrypted with the public key of their intended recipient, and a message stops flooding once a meter successfully decrypts it. Once all the proxies have received their values, the solution continues as in the prior approach. Notice that all nodes need to forward all messages. Thus with 100K nodes sending 100 byte messages, a query might trigger 10MB of traffic per participant. Although this is significantly more network load than our original scheme, it is feasible for a system with modern communications infrastructure, such as 4G wireless and broadband network backbones.

In practice we believe both methods might be useful. We would favor using the less costly scheme under normal conditions, but enabling the very redundant broadcast mode if

[1] We omit the proof here for brevity, but note that the function $g(i,j) = (i+1 \mod n, i + 2^j \mod n)$ induces an expander graph with diameter $\log n$.

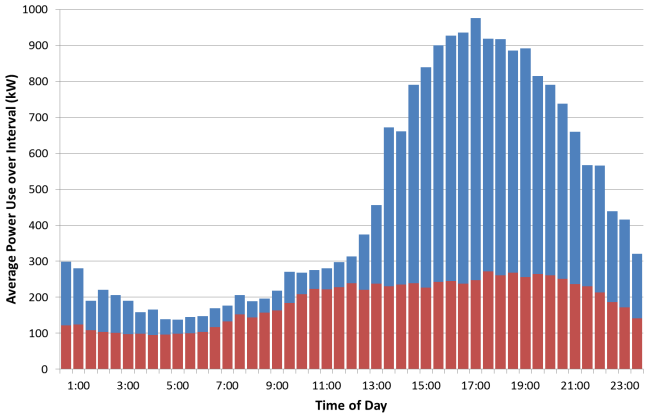

Fig. 1. Data collected by the utility, running queries using our system. Red bars are reported non-shiftable load, while blue bars are reported load from devices with DSM potential (i.e. heating and cooling systems).

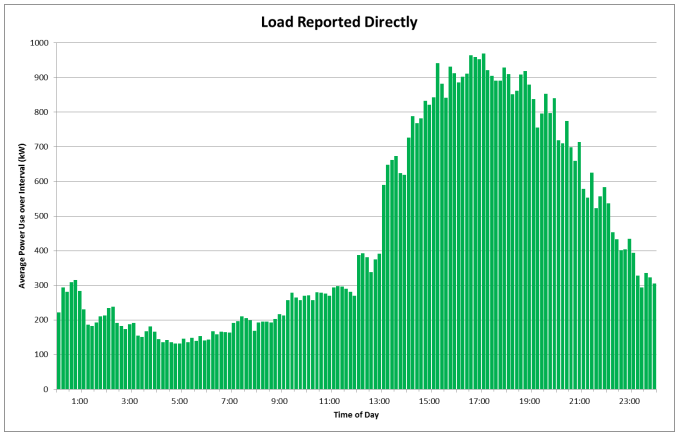

Fig. 2. Actual consumption recorded by meters, summed over all meters. Average power usage was obtained by dividing the energy usage recorded in one timestep (in kWh) by the timestep's length (1/6 of an hour).

major damage has occurred (for example after a storm) and the utility is still dealing with widespread failures.

D. Privacy and Fault-Tolerance

Here we will briefly explain why our system provides desirable privacy and fault-tolerance properties. For reasons of brevity we will not provided full proofs, but these can be found in our full-length paper [19].

Our algorithm protects the privacy of consumers in two ways: by anonymizing the values meters submit to a query, and by showing the utility only the results of the query rather than every individual contribution. The shuffle phase provides anonymity via the combination of random proxy choice and onion routing. A meter that receives a proxy value by decrypting the last layer of an onion cannot determine who sent it that value, since it does not know how many hops long the onion route was nor for how many rounds a previous intermediate meter might have held the onion before forwarding it. On the other hand, an intermediate meter between the source of a value and its proxy has no way of knowing what value it is forwarding, since it cannot decrypt further layers of the onion. In addition, the aggregate phase further hides the meters' contributions from the utility by computing the query in the network and sending only the result back to the utility. We require the query function to be commutative and associative so that each meter can compute and send a partial result using its local values and a previous result, instead of sending its local values individually to its parent in the tree. This ensures that no meter ever has the complete set of individual contributions (which might allow it to infer the identities of some contributions if it saw the complete set repeatedly), only a random subset of them.

Our algorithm produces the correct answer in the presence of up to t meter crashes due to its $t+1$ replication of the query computation. Since the meters send their values to proxies along independent (node-disjoint) overlay paths in the shuffle phase, any one meter could have at most t of its messages fail to reach a proxy due to failures (each failure interrupts only one path). Even in the worst case, one value reaches a proxy, and this proxy will send the value to all the other proxies in the echo phase. It is a property of our peer-to-peer overlay that the possible paths out of any one meter repeat only every n rounds of communication, so the paths used in the echo phase will be different than the paths used in the shuffle phase, rather than repeating the same forwarding paths and encountering the same failed meters. Thus, either t failures occur in the shuffle phase, or t failures occur in the echo phase, but not both. Once we have guaranteed that all $t+1$ proxies for a meter hold that meter's value, it is easy to see that at least one of the aggregation groups will produce the right query result, since they all start with the same set of values and at most t of them can contain failed meters.

IV. Simulation and Evaluation

We did not have access to smart meter hardware on which to test our algorithm. However, we implemented the algorithm in a software simulation of a smart grid. Our simulation uses a probabilistic model of electricity consumption, based on the one developed by Paatero and Lund in [21], to generate a realistic electrical load at each of n simulated homes. Each home has an associated "meter" that continuously records the electricity being consumed and can communicate with any other meter by sending a message through a simulated utility network. Although our simulation does not model the constraints of real meter hardware (all "meters" are memory objects on a single desktop computer, and communication is simply message passing in memory), it can be used to show our system's effectiveness at gathering meter information and its potential benefits when used for dynamic load balancing.

Paatero and Lund's model does not include home heating or cooling appliances, but these represent the largest source of electrical load in most households and present the best opportunity for demand-side load management via programmable thermostats. Therefore, we added central air conditioners, window air conditioners, and furnace fans to the model as possible

devices that could generate load at a home. We used data from the American Housing Survey [22] for the penetration rate of air conditioners and furnaces (determining how many simulated household had these devices), and information from several online datasheets (e.g. [23], [24]) for the per-cycle energy consumption of these devices.

Figure 1 shows the data that the simulated utility collected from our system over 24 hours, when running with 1019 simulated meters, by sending out queries every 30 minutes. The queries report two values, collected from every meter: the total energy consumption in kilowatt-hours since the previous query, and the total energy consumed by demand-side manageable devices since the previous query, representing load that is available for shifting. The utility divides these total values by the time interval being queried to obtain an estimate of the average power load, in kilowatts, from shiftable and non-shiftable devices over that time interval.

By comparison, figure 2 shows the data the utility could have obtained by continuously collecting all of the power consumption data recorded by every meter and computing the average power load in a centralized fashion. The intervals over which energy consumption is averaged are the smallest time scale measured by the meters, which is 10 minutes (the timestep in our simulation).

The close similarity in these figures shows that our protocol can provide the utility with data that is just as accurate as the data it could obtain by centralized, non-privacy-preserving monitoring. Note that the utility can approximate the "true" meter reading data arbitrarily well by increasing the frequency of queries. Since each meter needs to send only $2t \log n + 1$ messages to complete a single query, each query completes in approximately $2t \log n + 1$ network round-trips between the meters and the utility. With 100k meters and a network latency of 100ms, this means queries could run as frequently as every minute for values of t up to $\log n$.

V. Conclusion

Smart metering has the potential to greatly improve the efficiency of power systems by providing detailed, accurate, and timely information, but consumer concerns about the privacy of their data present a serious obstacle to the broad deployment of smart metering systems. Although there are many research proposals for privacy-preserving data collection, most existing solutions are either too computationally expensive to run on smart meters or limit the flexibility or accuracy of the queries that can be made. We have developed a data collection system for the smart grid that both protects consumers from surveillance and allows the utility company to gather accurate, useful data from smart meters. Our system uses only standard cryptography that should be easy to implement on smart meters, and can tolerate the transient failures that will occur in a large network of devices.

References

[1] "Advanced metering infrastructure (AMI)," Electric Power Research Institute, Palo Alto, CA, Tech. Rep. 1014793, Feb. 2007. [Online]. Available: http://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf

[2] Staff Report, "A national assessment of demand response potential," Federal Energy Regulatory Commission, Tech. Rep., Jun. 2009. [Online]. Available: http://www.ferc.gov/legal/staff-reports/06-09-demand-response.pdf

[3] G. P. Zachary, "Saving smart meters from a backlash," *IEEE Spectrum*, vol. 48, no. 8, pp. 8–8, Aug. 2011.

[4] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *IEEE Power and Energy Magazine*, vol. 1, no. 2, pp. 56–63, Mar. 2003.

[5] R. Anderson and S. Fuloria, "On the security economics of electricity metering," in *WEIS 2010*, Harvard University, Jun. 2010.

[6] X. He, M.-O. Pun, and C.-C. Kuo, "Secure and efficient cryptosystem for smart grid using homomorphic encryption," in *ISGT 2012*. Washington, DC: IEEE PES, Jan. 2012, pp. 1–8.

[7] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Security and Trust Management*, ser. LNCS, J. Cuellar, J. Lopez, G. Barthe, and A. Pretschner, Eds. Springer Berlin Heidelberg, Sep. 2010, no. 6710, pp. 226–238.

[8] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies*, ser. LNCS, S. Fischer-Hübner and N. Hopper, Eds. Springer Berlin Heidelberg, Jul. 2011, no. 6794, pp. 175–191.

[9] G. Ács and C. Castelluccia, "I Have a DREAM! (DiffeRentially privatE smArt Metering)," in *Information Hiding*, ser. LNCS, T. Filler, T. Pevný, S. Craver, and A. Ker, Eds. Springer Berlin Heidelberg, May 2011, no. 6958, pp. 118–132.

[10] I. Rouf, H. Mustafa, M. Xu, W. Xu, R. Miller, and M. Gruteser, "Neighborhood Watch: Security and privacy analysis of automatic meter reading systems," in *CCS '12*. Raleigh, NC: ACM, 2012, pp. 462–473.

[11] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE S&P*, vol. 7, no. 3, pp. 75–77, May 2009.

[12] M. Lisovich and S. Wicker, "Privacy concerns in upcoming residential and commercial demand-response systems," in *Proc. Clemson University Power Systems Conference*, Clemson University, Mar. 2008.

[13] Z. Erkin, J. Troncoso-Pastoriza, R. Lagendijk, and F. Perez-Gonzalez, "Privacy-preserving data aggregation in smart metering systems: An overview," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 75–86, Mar. 2013.

[14] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *SmartGridComm 2010*. IEEE, Oct. 2010, pp. 327–332.

[15] C. Rottondi, G. Verticale, and C. Krauss, "Distributed privacy-preserving aggregation of metering data in smart grids," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1342–1354, Jul. 2013.

[16] C. Dwork, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, no. 1, pp. 86–95, Jan. 2011.

[17] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *STOC '87*. New York, NY, USA: ACM, 1987, pp. 218–229.

[18] J. Saia and M. Zamani, "Recent results in scalable multi-party computation," in *SOFSEM 2015: Theory and Practice of Computer Science*, ser. LNCS, G. F. Italiano, T. Margaria-Steffen, J. Pokorný, J.-J. Quisquater, and R. Wattenhofer, Eds. Springer Berlin Heidelberg, Jan. 2015, no. 8939, pp. 24–44.

[19] E. Tremel, K. Birman, R. Kleinberg, and M. Jelasity, "Anonymous, fault-tolerant distributed data mining for smart devices," 2016, to appear.

[20] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[21] J. V. Paatero and P. D. Lund, "A model for generating household electricity load profiles," *International Journal of Energy Research*, vol. 30, no. 5, pp. 273–290, Apr. 2006.

[22] US Census Bureau, "National summary tables - AHS 2013," May 2015. [Online]. Available: http://www.census.gov/programs-surveys/ahs/data/2013/national-summary-report-and-tables---ahs-2013.html

[23] "How much electricity does my stuff use?" [Online]. Available: http://michaelbluejay.com/electricity/howmuch.html

[24] Lawrence Berkeley National Laboratory, "Default Energy Consumption of MELs." [Online]. Available: http://hes-documentation.lbl.gov/calculation-methodology/calculation-of-energy-consumption/major-appliances/miscellaneous-equipment-energy-consumption/default-energy-consumption-of-mels