Building a Secure and Privacy-Preserving Smart Grid

Ken Birman
Cornell University

Márk Jelasity
University of Szeged and
MTA-SZTE Research Group
on AI

Robert Kleinberg
Cornell University

Edward Tremel
Cornell University

ABSTRACT

New technologies for computerized metering and data collection in the electrical power grid promise to create a more efficient, cost-effective, and adaptable smart grid. 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. Our work proposes a design for a smart metering system that will allow utilities to use the collected data effectively while preserving the privacy of individual consumers.

1. INTRODUCTION

Smart grid technology has the potential to greatly improve the electrical grid. 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. A more informed and responsive electrical grid is also necessary to better integrate renewable sources into the energy supply, since they tend to generate power in a fluctuating manner that is incompatible with the slow-moving and rigid provisioning of the existing grid.

Unfortunately, the deployment of smart grid technology faces a serious obstacle in the form of concerns about customer privacy, which have already created vigorous opposition to smart meters [24]. 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) [16] 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 [2].

Our research addresses the problem of building a smart grid that protects consumer privacy while still realizing the benefits of computerized, fine-grained metering. In current and upcoming work, we will propose a design for a smart metering system that uses a combination of decentralization and differential privacy techniques to keep customers' private data hidden from the utility. We will show that our system still allows the utility to learn useful information about the electric grid and implement promising smart grid features such as load forecasting, accurate time-of-use billing, and demand-side management.

OSR This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM SIGOPS OSR 49(1) pp131–136, `http://dx.doi.org/10.1145/2723872.2723891`.
ACM X-XXXXX-XX-X/XX/XX...$15.00.

2. PROBLEM OVERVIEW

Before describing our proposal in detail, we will take a moment to clarify the assumptions we are making about the setup and environment of the system. The smart grid can be thought of as a large distributed system under a single administrative domain, namely that of the utility that owns or operates the grid. There are many client nodes, representing smart meters installed at individual homes or businesses, and a few servers under the direct control of the utility. This is an inherently centralized system, since all clients are connected to the utility's servers through a network set up and maintained by the utility (see Figure 1).

We assume that the networking hardware is reliable, so that any node in the network can send a message to any other node at will. Also, in order to help secure communications between nodes, we assume that the utility provides a membership server that clients can query for a reliable list of valid peers, as well as a standard PKI for certifying the public keys of client meters. We will expand on the necessity of this assumption in Section 3.

In order to model the concerns consumers have about the exposure of their private data, we treat the system operator (i.e. the utility) as an honest-but-curious adversary, which will run the system correctly but cannot be trusted with access to data it does not need to know. Although client systems will be owned and deployed by the system operator, public regulatory and oversight organizations routinely monitor and audit any power components deployed directly into the home. This creates a strong incentive for the utility to supply client devices that conform to specifications, rather than try and compromise them to achieve access to customers' private data.

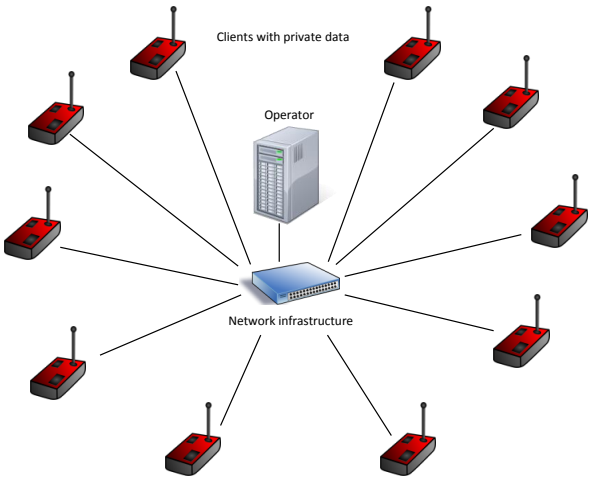

Figure 1: In the smart grid, a non-malicious but curious infrastructure owner provides a number of vital services: it connects clients with a routing network, maintains a membership service, and operates a PKI.

On the other hand, in any real-world system one must anticipate that Internet-connected personal computers will be subject to hacker and malware attacks. While we generally trust client systems with private data, we assume that the system may contain a small number of malicious, or Byzantine, nodes, fewer than $\log n$ where n is the total number of nodes in the system. Since the utility's membership service should enforce a policy of one identity per meter, the malicious nodes cannot use a Sybil attack [9] to artificially increase their numbers, but we still need to build a system that can tolerate a few malicious participants.

The goal of protecting consumers' privacy can be framed more concretely as a goal of preventing the utility from learning the underlying parameters of the probabilistic models that generate each home's meter readings. Each measurement recorded by a smart meter can be modeled as a random variable, and the distribution of this random variable depends partly on global parameters shared by all meters (e.g. the time of day, weather conditions, or public holidays) and partly on local parameters unique to each meter (e.g. the habits of the home's residents). After making a series of queries over these measurements, the utility should learn the query results but not the individual measurements or the local parameters that contributed to generating them.

In our work we make the simplifying assumption that distribution parameters are either fully local to an individual meter or common to all meters being queried. Although in reality there may be some factors that affect readings for a subset of meters, we do not consider the effects of such partially shared parameters.

As in many large distributed systems, we will organize our smart grid system in layers of abstraction that are fairly independent of one another. Of particular importance are the communication layer, which is responsible for connecting the meters and utility with a framework for distributing queries and responses, the data mining layer, which determines how data is collected and analyzed by the utility, and the control layer, which provides mechanisms for cooperative interaction between the meters and utility that directly affects power consumption. In the remainder of the paper, we will describe our work in each of these areas.

3. COMMUNICATIONS LAYER

One of our key design decisions in creating this system was that the data collected by smart meters should remain, as much as possible, on the meters themselves, rather than being uploaded to a central server. This means that most computation and aggregation of electricity consumption data happens at the edge, on the meters themselves, rather than within a utility-owned data center. We believe that keeping data on the consumer's device makes the problem of preserving privacy significantly more tractable, since we can then choose to only reveal to the utility the data that it needs to know instead of trying to prevent the utility from reading or analyzing data it already possesses. While there has been some work on preserving privacy in centralized data collection, it appears to be a much more difficult problem, and existing solutions require either computationally expensive cryptography (such as homomorphic encryption or secure multiparty computation [7]) or a restrictive model for how data can be aggregated [6].

The client-side computation approach is reminiscent of peer-to-peer systems, and in fact we create an overlay network that allows several existing peer-to-peer protocols to run across the meters. However, since we are designing for a system in which some nodes may be compromised or malicious, we chose not to adopt a purely peer-to-peer architecture because of its many security vulnerabilities. Determining the membership of a peer-to-peer network is hard, for example, since changes in membership may not be detected and propagated in a timely fashion [1], and malicious nodes can adopt multiple identities to masquerade as a large majority of the system [9]. On the other hand, a simple centralized membership server eliminates this problem, and this can be provided by the honest-but-curious infrastructure owner without compromising any private data from the clients. Thus in our communications layer, we combine peer-to-peer communications with some elements of centralized control, taking advantage of the existing centralized structure provided by the system operator.

In our upcoming paper [3], we present an overlay network that can support message exchange in a manner secure against many forms of manipulation or intrusion that might threaten privacy. We will briefly describe its setup here, though more details will be available in the paper.

Gossip protocols such as gossip-based aggregation [15] and distributed peer-to-peer learning [21] are a natural approach for analyzing data that remains on the client, and we wanted our system to support these existing solutions. However, the traditional random gossip algorithm that these protocols are based on, in which each node selects a random peer to exchange data with at regular intervals, can be extremely vulnerable to Byzantine participants. Since honest nodes have no way of determining whether their peers' choices are truly random, malicious nodes can send bogus gossip messages at a rapid rate to any or all of the other nodes, which must accept and process them if they are to correctly follow the protocol. We describe in our paper how this can lead to data corruption and denial-of-service attacks.

As a result, our communications system uses a completely deterministic algorithm to exchange gossip-like messages.

When combined with a PKI that allows client nodes to digitally sign their messages (and detect messages with invalid signatures), this allows honest nodes to quickly reject gossip messages that are not prescribed by the algorithm. Malicious nodes can thus be easily detected if they misbehave, and they can only affect the system at the same rate as the honest nodes, which are in the majority.

Our communication system works as follows. Each client node is assigned a unique integer ID between 0 and n, where n is the total number of nodes in the system. This assignment can be done by the utility's central server in any arbitrary fashion, since it already knows the identity of every meter connected to the network. Then, for each round j of gossip, each node i sends a message to the node with the ID determined by this function:

$$g(i, j) = i + 2^j \mod n$$

Before sending the message, the sending node signs it with its private key and encrypts it with the recipient's public key, to ensure that the message cannot be observed by the utility and the receiver will trust its validity. Since each node can independently calculate g for any ID, each node only accepts a message from the node that should be sending to it in the current round. Of course, the "rounds" do not need to be synchronous; we implement this system asynchronously, with each node maintaining its own round counter and using timeouts to automatically advance the round if the expected sender's message never arrives.

In order to ensure our deterministic system can replace random gossip as the basis for distributed computation, it should provide the same features as random gossip. Specifically, it should ensure that information can spread efficiently through the network (at least as quickly as random gossip, which converges in $O(\log n)$ rounds) and make nodes gossip as evenly as possible with the other nodes in the network, so that no one node's failure can become a bottleneck.

We designed our deterministic gossip function to be both perfectly efficient and perfectly uniform, provided the network size n is a prime number such that 2 is a primitive root modulo n.[1] Specifically, data that starts at any node and is forwarded upon receipt (as in standard epidemic gossip [8]) according to this function is guaranteed to reach all n nodes in $\lceil \log n \rceil$ rounds. Also, for any two nodes a and b, there is exactly one value of j in $[0, n-1]$ such that $g(a, j) = b$, which means that each node gossips with every other node exactly once before gossiping with the same node again. We formally prove these properties in our paper.

Figure 2 shows the pattern of data propagation in our system, visualized as if it was used to re-implement epidemic gossip. Note that the second set of $\log n$ rounds produces a different pattern of communication than the first set, so nodes do not repeat gossip partners in the entire n rounds.

Although the requirement for n to be a specific type of prime may seem like a difficult condition to meet, we experimentally determined that suitable values of n are sufficiently dense as to make it easy to find one that is very close to the actual network size. The unused node IDs that result from "rounding up" n to the nearest suitable prime can either be doubly assigned (giving a few nodes a second ID) or treated as failed nodes that immediately time out on all messages, which works surprisingly well in practice. Our experiments on an asynchronous implementation of gossip with our protocol show that it takes only a few extra rounds to converge when 10% of nodes have failed, and is hardly affected at all if those failures are known in advance (as they would be if the "failed nodes" are really nonexistent).

[1] In other words, the sequence 2, 4, 8, ..., 2^{n-1} cycles through each nonzero residue class modulo n exactly once.

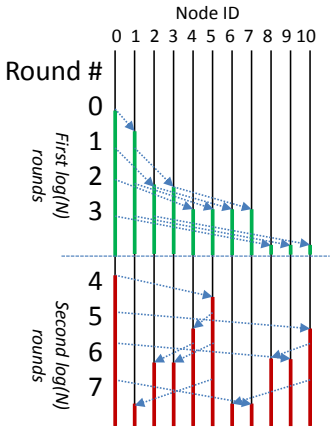

Figure 2: Information flow from node 0 in our scheme, showing two full epidemic cycles of $\log n$ rounds each. Every process sends and receives one message per round; we omitted the extra messages to reduce clutter. Similarly, although each round can be viewed as a new epidemic, the figure just shows two.

4. DATA MINING LAYER

Now that we have a robust and Byzantine-tolerant gossip system that the smart meters can use to communicate, we can begin to address the problem of collecting data from the meters in a way that preserves customer privacy. Assuming we use our overlay to exchange data among the meters without letting the utility observe any intermediate values, the goal is to ensure that the utility cannot determine the contribution of any single household by observing and analyzing the aggregate data set.

Following the framework of differential privacy introduced by Dwork in [10], we will use slightly noisy data to mask the contributions of individual meters' measurements to any aggregate reported to the utility. Each client should add noise to the meter readings it contributes to any aggregate being gossiped by drawing from a Laplacian distribution centered on the true value of the measurement. As Dwork explains in her work, properly calibrated Laplacian noise can prevent the utility from learning about the presence or absence of a single household with more than ϵ probability (where ϵ is a small, public parameter), while still keeping the data useful for making large-scale predictions.

However, standard differential privacy may not be enough to preserve customer privacy if we consider the fact that the utility will get to make many queries over data that come from relatively stable probability distributions. As we mentioned in Section 2, the measurements from each meter come from probability distributions whose parameters reflect personally-identifying characteristics of the household. Even if

each query the utility makes is independent from the others and the results hide individual contributions, the utility may be able to infer the underlying distribution parameters that generated these measurements once it has taken enough samples.

We explored this further complication, which we call distributional differential privacy, in [13]. In this work, we formally define the probabilistic model that generates a series of smart meter readings, and build a definition of distributional differential privacy from a definition of ordinary differential privacy. At a high level, we consider the series of readings taken by the set of smart meters to be a series of databases (one per unit of time), and the queries made by the utility to be an algorithm that returns a single output per input database. We assume these series to be infinite, in order to model the fact that privacy should be maintained no matter how many queries the utility makes. The algorithm is ϵ-differentially private if the probability of a given output changes by less than $\exp(\epsilon)$ when the input database is changed by a single record. In other words, the output of a query should not be significantly different with or without a single household's meter data.

We define distributional differential privacy in terms of distributional adjacency, which is based on the formalization of meter data as coming from probability distributions with household-specific parameters. Two series of databases are called distributionally adjacent if they contain the same set of readings except for one series of records, which is generated by a different distribution (i.e. a different hidden parameter) in the two series. This represents the situation where one household is replaced with another in a series of readings; the records from that household change, and the probability distribution they came from changes since each household has its own distribution parameter.

An algorithm is distributionally ϵ-differentially private if the probability of a given series of query outputs over a series of databases changes by less than $\exp(\epsilon)$ when the database series is replaced by a distributionally adjacent series. Intuitively, this means that the results of any queries on a series of readings should not change significantly if exactly one of the hidden distribution parameters changes. So the utility could not detect the replacement of one household with another (and thus cannot infer a single household's distribution) no matter how many queries it makes.

Note that this is a rather pessimistic definition of privacy, since it assumes that the public, global parameters that affect the distribution of each household's readings remain constant over an infinite number of readings. Since these parameters actually represent factors such as level of daylight and weather conditions, they are likely to change within the time that the utility could be making queries. In that case, the effect of changing public parameters (and hence a changing distribution) on the query results could mask the effect of a single household's contribution, and the utility might be unable to detect that household's presence even if K was not differentially private by this definition.

This definition also assumes that the private, per-household distribution parameters are constant, which is more realistic since they represent the usage habits of the household's occupants. Nonetheless, it is still a worst-case assumption, because over an infinite number of queries even "static characteristics" will change, if the occupants change their lifestyle or move away. Such changes to the hidden parameters during the course of the utility's queries would make it more difficult for the utility to infer their value.

In our work we explore a few ways to achieve distributional differential privacy, even for this pessimistic definition. It turns out that if the meters' readings come from Gaussian distributions, the same noise-generation techniques that are commonly used to achieve differential privacy can be extended to hide distributional parameters. Namely, a noise variable drawn from a Laplacian distribution with the same mean as the underlying meter distribution is added to the query result. Although in reality the readings' distributions will not be Gaussian, we show how the meters can transform their readings to Gaussian distributions before reporting them, so that the Laplacian noise will be effective at preserving distributional privacy.

The practice of adding noise to meter readings in order to preserve privacy seems problematic when it comes to billing customers for electricity usage. Customers would be unhappy if they were billed based on intentionally-inaccurate meter readings, but showing the utility accurate data at billing time defeats the purpose of adding noise during other queries. Fortunately, the problem of generating a bill while preserving privacy is smaller in scope than privacy-preserving data mining, and it has been well-studied.

For example, Rial and Danezis [23] show how exact readings can be used to create an exact bill, and send that bill to the utility, without revealing the readings to the utility. Their protocol uses zero-knowledge proofs and commitments to guarantee to the utility that the meter faithfully applied the utility-supplied pricing function to its readings. While this adds a significant amount of cryptographic overhead, and might not be applicable to general-purpose data mining, it is practical and effective for a once-per-month bill generating operation.

Thus in our proposed system, differential privacy noise is only applied to meter readings that are collected for data mining purposes, such as predicting load, that do not affect a customer's bill. Bill generation will use the original pre-noise readings, and we will employ a system similar to Rial and Danezis's to ensure that the utility does not learn the readings during billing.

5. CONTROL LAYER

We see our smart meter communication and data aggregation system as part of a broader ecosystem of cooperative control involving the utility company and its customers. Both parties have an incentive to cooperate in using the data collected by smart meters to implement demand-side management of power usage. The utility can use data supplied by the meters to predict demand far enough in advance to avoid expensive last-minute purchases of electricity, thus reducing its operating costs. If clients can also adjust their demand in response to requests by the utility, the utility can smooth out power demand to more closely match capacity, or adjust it to follow fluctuations in power from renewable sources. Meanwhile, customers can benefit by saving money on electricity (assuming the price of energy reflects its cost to the utility) and receiving better protection from brownouts. Customers that want to be environmentally friendly would also encourage the utility to use renewable power by participating in cooperative control.

The most basic way in which our communication and data mining layers can be used to help manage the grid is by

predicting load. Currently, most electric utility companies determine when to schedule power generation based on general historical trends for the region at the given time of year, but during the course of a day power demand inevitably differs from this coarse-grained prediction. When demand is lower than expected electricity is wasted, and when demand is higher than expected electricity must be purchased from other utility companies at great expense, because most power plants take at least four hours to be activated or deactivated. By using our privacy-preserving data aggregation system, however, the utility could regularly ask the meters to build a machine learning model predicting their usage for the next several hours. This model would be more accurate than a prediction from general historical trends (though it could use historical trends as a prior) because it would incorporate the specific usage data being recorded that day, and it would give the utility enough advance notice of higher-than-usual demand to activate additional power sources.

A smart grid system such as ours can achieve even more benefits if customers have smart devices, appliances that can wirelessly connect to a smart meter and be programmed to turn on at a time scheduled by the meter. These are usually appliances such as dishwashers, laundry machines, or hot water heaters that can be run any time within a range of hours without inconveniencing the user. With this setup, the utility can use our communications system to advise such devices on when to run, and the cumulative effect of scheduling all the appliances within a region can reshape energy demand. One possible implementation is for the utility to send out a "schedule" at the beginning of each day, indicating its desired total demand for each hourly interval and the increase in energy price that will result if demand crosses this threshold. The meters would then communicate with each other in a peer-to-peer fashion to agree on when they will run each of their devices, possibly using an auction-like bidding system or a cost-minimization algorithm (such as the one proposed by Mohsenian-Rad et al. in [20]). Since the meters would use our secure overlay for communication, the utility would only learn the final overall schedule (which it could use as a reliable prediction of demand), not the times any individual household will be running its appliances.

Although most work on smart grid systems designs for the situation in which the entire grid has the same smart features, it is important to remember that the rollout of smart meters and smart devices will take several years. During that period, only some communities or neighborhoods within a utility's region of service will be connected to the smart grid network. The question then arises whether the utility can still derive the promised benefits of a smart grid while there is only partial penetration of the technology. In order to answer this question, and determine what the minimal level of rollout is to see noticeable benefits, we have implemented a software simulation of a power grid using the probabilistic models of household demand created by Paatero and Lund in [22]. Our preliminary results are inconclusive, but this is still a work in progress.

Another caveat is that both the data aggregation and demand response layers will need to take into account compromised or malicious nodes, like the communications layer does. We have a solution in hand for Byzantine-tolerant aggregation of data, and should be able to implement a similar solution for the control system.

6. RELATED WORK

The privacy problems related to smart metering have been studied before. McDaniel and McLaughlin [19] surveyed many of the security and privacy concerns that can arise in smart grids, and Lisovich and Wicker [18] identify many of the concrete technical challenges to preserving privacy.

Our communications layer is based on years of work in gossip protocols, starting with Demers et al.'s work in [8]. Bimodal multicast [4] and LPBcast [12] are two notable gossip multicast systems that inspired our overlay network's design. The problem of malicious participants in gossip networks has been studied extensively, and systems such as Brahms [5] and Secure Peer Sampling [14] use a purely peer-to-peer approach to stop Byzantine nodes from poisoning honest nodes' view of system membership. BAR Gossip [17] is one of the earliest gossip systems that tolerates malicious nodes. The authors assume that gossip is being used specifically to deliver a streaming broadcast from some origin node, and they propose a model in which some nodes are Byzantine while others are "rational adversaries" that will attempt to receive as much of the streaming broadcast as possible without doing their part to transmit it. They use a similar approach to ours to limit the damage that can be done by Byzantine nodes, namely requiring digital signatures on messages and using a protocol that limits the rate at which nodes can send legitimate messages.

There has been much prior work on privacy-preserving data mining from the cryptography community. The most similar to our work is the algorithm developed by Dwork et al. in [11]. It uses a secret sharing scheme to allow a distributed system of participants to mix measurements and noise in an oblivious fashion, such that the final result can only be viewed once it can hide any participant's contribution. This system works well, but we wanted to develop a simpler system with less cryptographic overhead that could still provide the privacy needed for the smart grid.

7. CONCLUSION

Widespread deployment of smart grid technology has the potential to revolutionize the electric power system, but smart grids create risks because electricity is so closely integrated with our personal lives. Power utilities need to carefully consider the privacy and security problems associated with a proposed smart grid system before deploying it if they are to avoid serious system compromises and consumer outrage over intrusive surveillance. We provide a road map for building a privacy-preserving smart metering system that can be used for data collection, load prediction, and cooperative demand management. Rather than collect all smart meter data in a central location, our design keeps data on the meters themselves and uploads only aggregate data to the utility. We use differential privacy techniques to ensure that the aggregated data cannot reveal the contribution of an individual meter, and use Byzantine fault-tolerant algorithms to ensure that our client-focused system is not vulnerable to a few hacked or malicious meters.

Acknowledgements

This work was supported, in part, by grants from the National Science Foundation "Smart Grids" program and by the DOE Advanced Research Projects for Energy (ARPA-e) GENI program.

8. REFERENCES

[1] André Allavena, Alan Demers, and John E. Hopcroft. Correctness of a gossip based membership protocol. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Principles of Distributed Computing*, PODC '05, pages 292–301, New York, NY, USA, 2005. ACM.

[2] Ross Anderson and Shailendra Fuloria. On the security economics of electricity metering. In *The Ninth Workshop on the Economics of Information Security (WEIS 2010)*, Harvard University, 2010. Citeseer.

[3] Ken Birman, Márk Jelasity, Robert Kleinberg, and Edward Tremel. A secure communications overlay for privacy-preserving distributed computation. 2014. Submitted for publication. Preliminary version available upon request.

[4] Kenneth P. Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. *ACM Trans. Comput. Syst.*, 17(2):41–88, 1999.

[5] Edward Bortnikov, Maxim Gurevich, Idit Keidar, Gabriel Kliot, and Alexander Shraer. Brahms: Byzantine resilient random membership sampling. In *Proc. 27th ACM Symp. on Principles of Distributed Computing*, PODC '08, page 145–154. ACM, 2008.

[6] Ruichuan Chen, Istemi Ekin Akkus, and Paul Francis. SplitX: High-performance private analytics. In *Proc. ACM SIGCOMM*, page 315–326. ACM, 2013.

[7] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2):28–34, December 2002.

[8] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proc. 6th Annual ACM Symp. on Principles of Distributed Computing*, PODC '87, page 1âĂŞ12. ACM, 1987.

[9] John R. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, number 2429 in LNCS, pages 251–260. Springer, 2002.

[10] Cynthia Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95, January 2011.

[11] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, number 4004 in Lecture Notes in Computer Science, pages 486–503. Springer Berlin Heidelberg, January 2006.

[12] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.*, 21:341–374, 2003.

[13] Márk Jelasity and Kenneth P. Birman. Distributional differential privacy for large-scale smart metering. In *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '14, pages 141–146, New York, NY, USA, 2014. ACM.

[14] Gian Paolo Jesi, Alberto Montresor, and Maarten van Steen. Secure peer sampling. *Computer Networks*, 54(12):2086–2098, 2010.

[15] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symp. on Foundations of Comp. Science*, pages 482–491, 2003.

[16] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 1(2):56–63, March 2003.

[17] Harry C. Li, Allen Clement, Edmund L. Wong, Jeff Napper, Indrajit Roy, Lorenzo Alvisi, and Michael Dahlin. BAR gossip. In *Proc. 7th Symp. on Operating Systems Design and Implementation*, OSDI '06, pages 191–204. USENIX Association, 2006.

[18] Mikhail Lisovich and Stephen Wicker. Privacy concerns in upcoming residential and commercial demand-response systems. In *Proc. of the Clemson University Power Systems Conference*, Clemson University, March 2008. Citeseer.

[19] Patrick McDaniel and Stephen McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7(3):75–77, May 2009.

[20] Amir-Hamed Mohsenian-Rad, Vincent W.S. Wong, Juri Jatskevich, Robert Schober, and Alberto Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Transactions on Smart Grid*, 1(3):320–331, December 2010.

[21] Róbert Ormándi, István Hegedűs, and Márk Jelasity. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 25(4):556–571, 2013.

[22] Jukka V. Paatero and Peter D. Lund. A model for generating household electricity load profiles. *International Journal of Energy Research*, 30(5):273–290, April 2006.

[23] Alfredo Rial and George Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, page 49–60, New York, NY, USA, 2011. ACM.

[24] G. Pascal Zachary. Saving smart meters from a backlash. *IEEE Spectrum*, 48(8):8–8, August 2011.