Proactive Fault-Tolerant Aggregation Protocol for Privacy-Assured Smart Metering

Jongho Won[◇] Chris Y.T. Ma[♯] David K.Y. Yau[†♯] Nageswara S.V. Rao[□]
[◇] Purdue University, West Lafayette, IN, USA
[♯] Advanced Digital Sciences Center, Singapore
[†] Singapore University of Technology and Design, Singapore
[□] Oak Ridge National Laboratory, TN, USA
Email: won12@cs.purdue.edu, chris.ma@adsc.com.sg, david_yau@sutd.edu.sg, raons@ornl.gov

***Abstract*—Smart meters are integral to demand response in emerging smart grids, by reporting the electricity consumption of users to serve application needs. But reporting real-time usage information for individual households raises privacy concerns. Existing techniques to guarantee differential privacy (DP) of smart meter users either are not fault tolerant or achieve (possibly partial) fault tolerance at high communication overheads. In this paper, we propose a fault-tolerant protocol for smart metering that can handle general communication failures while ensuring DP with significantly improved efficiency and lower errors compared with the state of the art. Our protocol handles fail-stop faults proactively by using a novel design of *future ciphertexts*, and distributes trust among the smart meters by sharing secret keys among them. We prove the DP properties of our protocol and analyze its advantages in fault tolerance, accuracy, and communication efficiency relative to competing techniques. We illustrate our analysis by simulations driven by real-world traces of electricity consumption.**

I. Introduction and Related Work

The traditional power grid has a *supply-follows-demand* usage model. Demand-response (DR) in emerging smart grids holds promise for a *demand-follows-supply* alternative [2], which has broad implications including (i) economically-efficient electricity consumption in which elastic demand at peak times is shifted to off-peak periods when supply is much less expensive, and (ii) integration of intermittent green energy sources (such as solar and wind) by managing demand on-line to match fluctuating supply levels. By reporting essential load information for key control decisions (e.g., how much load to shed or shift when and where), smart metering is a critical enabling technology for DR to succeed.

Smart meters report consumption for users at high frequency (e.g., once per second) and in real time. This level of monitoring can reveal much private information about the users and subject them to various undesirable outcomes [10], [14], e.g., whether they use the exercise room much (discriminating pricing of health insurance), if they watch TV a lot (predatory advertising), or even stealthy surveillance in general [7]. Public outcry about privacy has led to the banning of smart meters in North American cities [19], and derailed a planned mandatory deployment of smart meters in the Netherlands [8]. Where they are still deployed, users must now consent to opting in voluntarily. It is clear that users will not opt in if the privacy implications of doing so remain unclear.

In this paper, we aim to remove critical privacy barriers for users to participate in DR, so that current advances in smart grid technologies can fulfill their true promise of economic and social impact. Our specific aim is to provide strong privacy assurance for users who contribute their consumption data to aggregators managing DR programs. We adopt *differential privacy* (DP) [4] as the privacy notion because as an information-theoretic measure, it is independent of any attack methods or indeed any assumed computational limitations of the adversary. It is also a strong notion that guarantees the privacy of any single value in a dataset even if the adversary knows all the other values.

DP is a privacy approach of much current interest [5], [11] and its use in DR aggregation protocols is mainly characterized by the need to distribute the noise for privacy protection among smart meters which report to the aggregator [1], [3], [15], [18]. Certain of the prior aggregation protocols are not fault tolerant [15], [18], meaning that they will fail if some participating meter(s) fails to report. This presents clear difficulties in practical deployments, since smart meters, as inexpensive home devices in unprotected environments, are unreliable, and their communication typically occurs over unreliable network channels as well. It is unacceptable that the unavailability of isolated parts of the network will prevent the operation of the DR program as a whole.

To achieve fault tolerance in aggregation protocols, there are two possible approaches: *reactive* and *proactive*. A reactive protocol learns about failures after-the-fact and initiates recovery actions from them afterwards. A state-of-the-art reactive protocol [1] achieves small errors in spite of failures, but it requires multiple rounds of message exchanges between the smart meters and aggregator, which increases the required network bandwidth and delay. More importantly, the protocol is tolerant of *partial* failures only.

We now give a succinct account of how the reactive protocol in [1] can fail under communication failures. A basis of the protocol is that it distributes noise for privacy and keys for message encryption among the meters, which report their respective shares to the aggregator. When the aggregator sums up the reported values, the noise should add up to that required for differential privacy, and the keys *must* sum to zero. For fault tolerance, the meters have set up prior partnership agreements with each other, so that two partners share each other's keys.

The protocol works as follows (see [1] for details). In round one, the meters report. In round two, the aggregator broadcasts to all the meters an id list of the meters who did not report (i.e., failed). In round three, a working meter, say i, who is a partner of a failed meter, say j, reports j's key on behalf of j. It is clear that if either round two or three fails, the keys do not cancel out and the aggregation fails.

In a proactive protocol, smart meters send redundant information in anticipation of faults. Should a fault occur actually, the needed information for recovery is already available, which reduces the recovery time significantly. Indeed, a proactive protocol may handle failures using one (unidirectional) communication step from the working meters to the aggregator only [3].

The state-of-the-art proactive protocol, the binary protocol [3], suffers from several important practical problems, however. First, the binary protocol sends redundant information for proactive recovery as a binary interval tree encoding of the meters, which has $O(\log N)$ bandwidth cost even in normal operation (i.e., no failures). Hence, it achieves low delay at the expense of high bandwidth (if $N = 4000$, the bandwidth cost is 12 times). Second, its bandwidth cost for supporting meter join/leave is linear in N. Since N can be quite large in practice, the overhead is of concern in dynamic environments where there are non-negligible churns of participants for reasons such as meters turned on and off, plugged in and unplugged, or failing and recovering. Third, the error of the protocol grows with N as well (see Table I). A large error will compromise the effectiveness of the demand response. For example, if an aggregator uses inexact aggregate consumption to determine real-time electricity prices, the prices may not be fair or truly market-efficient.

In this paper, our **main contribution** is the design, privacy analysis, error analysis, and complexity analysis of a proactive fault-tolerant aggregation protocol for privacy-assured demand-response.

- Compared with the state of the art in reactive fault tolerance [1], our protocol is significantly more efficient in supporting meter join. Although it sends redundant information for proactive recovery, in steady-state normal operation (i.e., no failures), the redundancy is only one *future ciphertext* (see Sec. IV-C), whose presence doubles the message size, per time slot. In comparison, the reactive protocol does not increase the message size, but it requires three messages per slot, compared with one in the proposed protocol. Hence, our protocol is *more bandwidth-efficient, by about 50%*. More importantly, our protocol is fully resilient against communication failures, whereas the protocol in [1] is not as we discussed.
- Compared with a state-of-the-art binary protocol [3], our protocol has the same level of fault tolerance. However, our protocol (i) is significantly more efficient in normal operation (constant bandwidth vs. $\log N$), (ii) is significantly more efficient in supporting meter join/leave (constant bandwidth vs. linear in N), and (iii) has significantly reduced error which grows with the number of failed meters w only, but not N (Table I).

Table I compares our protocol with major related protocols in the literature, in terms of fault tolerance, communication complexity (bandwidth cost), communication model, and error.

Our **other contributions** are as follows. We present how the division of a privacy budget in our protocol can be optimized to minimize the aggregation error (i.e., the RMSE defined in Sec. IV-E), and show how the error can be reduced further by a notion of *individual sensitivity*. Furthermore, we present simulation results based on real-world traces of electricity consumption in the U.K. and Singapore to illustrate the performance of the proposed protocol and its improvements over the binary protocol in real-world environments.

II. Preliminaries

We adopt differential privacy (DP) [4] as our privacy definition. For background, we discuss its meaning, noise generation techniques, and composition properties.

A. Differential privacy

Definition 1. (ϵ-differential privacy) *A randomized algorithm $\mathcal{A}$ is* ϵ-differentially private *if for any two datasets D_1 and D_2 that differ on a single element, and for all $S \subseteq Range(\mathcal{A})$,*

$$Pr(\mathcal{A}(D_1) \in S) \leq exp(\epsilon) \cdot Pr(\mathcal{A}(D_2) \in S).$$

A differentially private algorithm $\mathcal{A}$ provides privacy because, given any two datasets which differ on a single element only, respective results of a same query on the datasets are not distinguishable. Therefore, an adversary cannot infer the value of any single element in the dataset. Here, ϵ represents the level of privacy. A smaller value of ϵ means better privacy, but it also implies lower accuracy of the query result.

B. Differential privacy via Laplace noise

Dwork [4] proves that adding i.i.d. Laplace noise $Lap(\lambda)$ to a query result achieves ϵ-differential privacy. The Laplace noise $Lap(\lambda)$ is sampled from a Laplace distribution of parameter λ defined as

$$\lambda = \frac{GS_f}{\epsilon},$$

where GS_f denotes the *global sensitivity* of the function f. In smart metering, f is the electricity consumption of say a household, so GS_f is the maximum amount that any one household can consume over any reporting period. λ is also called the *noise scale*. A random variable that follows the Laplace distribution has standard deviation $\sqrt{2} \cdot \lambda$ and expected absolute deviation λ. As a result, the smaller ϵ is, the noisier the outputs and hence the higher the level of privacy guaranteed.

C. Distributed Laplacian noise generation

In demand-response (DR) smart metering, if there are adversaries against the privacy of participating users, each meter should generate shares of random noise in a distributed manner. Dwork *et al.* [6] show that Gaussian noise provides (ϵ, δ)-DP, where δ is the probability that the loss of privacy is not bounded by ϵ. In [18], each meter adds Laplace noise probabilistically to achieve (ϵ, δ)-DP. Alternatively, [1] makes use of the infinite divisibility of the Laplace distribution to achieve ϵ-DP. It is

TABLE I: Comparison between the proposed protocols and related aggregation protocols for smart metering.

Scheme	Fault-tolerant?	Bandwidth in normal operation	Bandwidth for join, leave	Comm. model in reporting	Error (RMSE)	Encryption
[15]	No	$O(1)$	$O(1), O(1)$	C $\Leftrightarrow$ S	$O(1)$	Homomorphic
[18]	No	$O(1)$	$O(N), O(N)$	C $\rightarrow$ S	$O(1)$	Exponentiation
[1]	Partially	$O(1)$	$O(N), O(1)$	C $\Leftrightarrow$ S	$O(1)$	Modular addition
[3]	Yes	$O(\log N)$	$O(N), O(N)$	C $\rightarrow$ S	$\tilde{O}\left((\log N)^{1.5}\sqrt{w+1}\right)$	Exponentiation
This paper	Yes	$O(1)$	$O(k \times B), O(k)$	C $\rightarrow$ S	$O(\sqrt{w+1})$	Modular addition

N is the total number of meters and w is the number of failed meters. The $\tilde{O}(\cdot)$ notation hides a $\log \log N$ factor. C $\Leftrightarrow$ S means bi-directional communication between client and server, C $\rightarrow$ S means client-to-server uni-directional communication. k and B are constant design parameters of the protocol; $k \ll N$ and $B \ll N$ in practice. Our scheme achieves true fault tolerance against communication failures, unlike [1], [15], [18]. It has the same fault tolerance as [3], but much reduced errors and bandwidth overheads (see Sec. IV-C for details). It also requires uni-directional communication only when smart meters report to the aggregator. Our scheme uses modular addition-based encryption, which is much more efficient than homomorphic or exponentiation-based encryption.

known that the Laplace distribution can be assembled from the sum of i.i.d. gamma distributions [9].

Infinite divisibility of Laplace distribution [9, Proposition 2.4.1] *Let $Lap(\lambda)$ denote a random variable which is sampled from a Laplace distribution with pdf $f(x,\lambda) = \frac{1}{2\lambda}e^{|x|/\lambda}$. Then the distribution $Lap(\lambda)$ is infinitely divisible. Furthermore, for every integer $n \geq 1, Lap(\lambda) = \sum_{i=1}^{n}(G(n,\lambda) - G'(n,\lambda))$, where $G(n,\lambda)$ and $G'(n,\lambda)$ are i.i.d. with gamma density $g(x,n,\lambda)$. The gamma density is defined as*

$$g(x,n,\lambda) = \frac{(1/\lambda)^{1/n}}{\Gamma(1/n)} x^{\frac{1}{n}-1} e^{-x/\lambda},$$

where $\Gamma(1/n)$ is the Gamma function evaluated at $1/n$.

In our case, if the number of smart meters in an aggregation group is N, each smart meter, say i, adds $G_i(N,\lambda) - G'_i(N,\lambda)$ to its consumption data x_i before reporting. Then the sum of the reported data is given by

$$\sum_{i=1}^{N} x_i + \sum_{i=1}^{N}(G_i(N,\lambda) - G'_i(N,\lambda)) = \sum_{i=1}^{N} x_i + Lap(\lambda).$$

Therefore, ϵ-differential privacy is provided.

D. Composability

The composition of differentially private algorithms also provides differential privacy, but it produces different results depending on the data to which the queries are applied.

Sequential composition [12, Theorem 3] *Let $\mathcal{A}_i$ each provide ϵ_i-differential privacy. The sequence of $\mathcal{A}_i(X)$ provides $(\sum_i \epsilon_i)$-differential privacy.*

Parallel composition [12, Theorem 4] *Let $\mathcal{A}_i$ each provide ϵ_i-differential privacy. Let D_i be arbitrary disjoint subsets of the input domain D. The sequence of $\mathcal{A}_i(X \cap D_i)$ provides ϵ-differential privacy.*

In other words, if we run an ϵ-DP algorithm t times on a dataset X, the result becomes $\epsilon \times t$-differentially private. However, if we run an ϵ-DP algorithm t times on disjoint subsets of the dataset X, the result remains ϵ-differentially private.

III. System Model and Problem Definition

A. Problem definition

We assume that in a DR deployment, there are one aggregator and N smart meters. Each smart meter, say i, measures its electricity consumption $x_{i,t}$ in each time slot t, and sends it to the aggregator at the end of the time slot. The aggregator only needs to know $\sum_{i=1}^{N} x_{i,t}$, i.e., the total power consumption in t, in order to control the demand response, e.g., reduce peak loads of the power grid or match the aggregate consumption to available supply. The aggregator does not need to know the electricity consumption of individual users, and we aim to protect this private information from possible privacy attacks by a "curious" aggregator. To do so, we need to design a differentially private protocol to aggregate the electricity consumption reported by individual smart meters in real time. We seek the following desirable features for the protocol.

- **Privacy protection with small errors.** The protocol should allow the aggregator to know the total electricity consumption of users with little loss of accuracy to ensure the performance of the demand response, while protecting the privacy of individual consumption data.
- **Cost effectiveness.** Smart meters are home devices that must have low cost, either for consumer adoption or for utility companies to pay for them at large scales. As such, their computing power is limited. Therefore, the protocol should have low computational requirements at the meters. Similarly, the protocol should have low communication overheads and bandwidth requirements, to reduce the cost of the supporting network infrastructure.
- **Scalability.** The communication overhead of the protocol should remain small even if the number of participating smart meters is big in large-scale deployments.
- **Fault tolerance.** As low-cost devices running in unprotected environments, smart meters are prone to failures. Transient or longer-term communication failures are also not uncommon, due to say network congestions, routing changes, and faulty network connections. Therefore, the protocol should be tolerant of device and communication failures (fail-stop faults), so that the DR program may remain operational as a whole even if parts of the infrastructure should become unavailable.

B. Trust model

In this paper, we consider a semi-honest (honest-but-curious) trust model [1], [3], [15], [18]. The aggregator is untrusted in that a *curious aggregator* may try to compromise someone's private information through the aggregation protocol. A small fraction of the smart meters may collude with the curious aggregator. Smart meters will in general hide the information they have before reporting it to the aggregator. To assist the curious aggregator, however, colluders may deviate from the protocol by providing their own information in the clear to the aggregator. In general, the aggregator will follow the prescribed protocol, except that a curious aggregator may try to exploit illegitimate information provided by colluders.

C. Network model

We assume that smart meters have a bi-directional communication channel with the aggregator. The smart meters are not connected to each other directly, but they can exchange encrypted messages among themselves via the aggregator or intermediate routers through end-to-end secure channels. Although the communication links between smart meters and the aggregator are fairly reliable, loss of communication may still happen occasionally, due to reasons such as network congestions and routing changes. Of course, we may also lose communication if the source of the communication, i.e., a smart meter, fails. We refer to the loss of communication as a *communication failure*, whether the problem is with the meter or the communication channel.

IV. Protocol Description and Analysis

A. Initial setup

Each smart meter is configured with a private key and embedded with a corresponding certificate issued by a certification authority. The certificates are issued by trusted manufacturers when they produce the meters. When a new meter joins the network, it sends its certificate to the aggregator. The aggregator assigns identification numbers sequentially for all the smart meters as they join. After registering a new meter that joined, the aggregator broadcasts the new meter's id, as well as the (possibly changed) maximum id N, so that every meter knows N. We assume that meter ids are reused once their old owners left the system and their associated keys expired, so that N is approximately equal to the total number of smart meters participating in the protocol.

After a new meter, say i, joins, it randomly selects k other meters as *partners*, and requests their certificates (public keys) from the aggregator. Here, k is a system parameter that should be large enough so that the probability that all the k partners of i are colluders is negligible. We assume that secure channels between partnering smart meters are established based on public key encryption. Hence, for each partner meter, say m, i chooses a random number $sk_{i,m}$ to be the encryption key that it shares with m, and sends the shared key to m.

To avoid the case that many meters select (randomly) the same meter m as partner and hence share secret keys with it, which strains m's local memory to store the keys, each meter will only accept shared keys from at most $k + C$ other meters, where C is a small constant and $k \gg C$. After this limit is reached, the meter will refuse to accept more partners. In general, meter i will try to select another meter as a new partner if (i) i sent a partner request to an original meter m but either m refuses or i does not get an acknowledgment from m, or (ii) an existing partner of i's, say m, leaves the system (see Sec. IV-C).

B. Secure aggregation without fault tolerance

In this section we describe how a meter reports its readings to the aggregator in a secure way. As an initial step, a smart meter, say i, calculates $\tilde{x}_{i,t} = x_{i,t} + G_{i,t}(N, \lambda) - G'_{i,t}(N, \lambda)$, where $x_{i,t}$ is its consumption in time slot t, $G(N, \lambda)$ and $G'(N, \lambda)$ are i.i.d. random variables with gamma density, and the sum of the random variables from all the meters guarantees differential privacy, due to the infinite divisibility of the Laplace distribution stated in Sec. II-C.

However, $\tilde{x}_{i,t}$ by itself is not enough to ensure the privacy of meter i because the noise of $G_{i,t}(N, \lambda) - G'_{i,t}(N, \lambda)$ may not perturb $x_{i,t}$ sufficiently [1]. As a result, $\tilde{x}_{i,t}$ should be encrypted as a *ciphertext* before being sent to the aggregator – the aggregator can only decrypt the sum of ciphertexts from all the meters but not the individual ciphertexts. We adopt modular addition-based encryption. This is because a smart meter has limited computing power and modular addition-based encryption is significantly faster than exponentiation-based encryption. Moreover, modular addition-based encryption produces smaller ciphertexts in general.

In the encryption step, a meter, say i, encrypts $\tilde{x}_{i,t}$ by adding a random number $r_{i,t}$. The random number is formed from component random numbers generated by a secure pseudo random number generator ($PRNG$) with the meter's shared keys as seeds. Specifically,

$$r_{i,t} = \sum_{m \in M} PRNG(sk_{i,m}, t) - \sum_{l \in L} PRNG(sk_{l,i}, t),$$

where M denotes the set of partners i chose and L denotes the set of partners that chose i. Note that the encryption uses the sum operation because it is addition-based. Also, t is a global and public time slot number that increases after every reporting time interval of consumption. Since t changes at every time slot, $r_{i,t}$ also changes at every time slot. Note that $\sum_{i=1}^{N} r_{i,t} = 0$ since $PRNG(sk_{i,m}, t)$ in $r_{i,t}$ is equal to $PRNG(sk_{i,m}, t)$ in $r_{m,t}$ and they cancel out in the summation performed by the aggregator. Smart meter i then sends $\tilde{x}_{i,t} + r_{i,t}$ as a ciphertext to the aggregator.

If the aggregator receives all the ciphertexts, it can calculate the aggregated sum as follows:

$$\sum_{i=1}^{N} (\tilde{x}_{i,t} + r_{i,t}) = \sum_{i=1}^{N} \tilde{x}_{i,t} + \sum_{i=1}^{N} r_{i,t} = \sum_{i=1}^{N} x_{i,t} + Lap_t(\lambda).$$

This protocol satisfies ϵ-differential privacy due to $Lap_t(\lambda)$. If we assume that a fraction p_c of the smart meters are colluders in that they supply clear information without adding the prescribed Gamma noise, then each smart meter adopts $G_{i,t}(N - N \cdot p_c, \lambda) - G'_{i,t}(N - N \cdot p_c, \lambda)$ instead of $G_{i,t}(N, \lambda) - G'_{i,t}(N, \lambda)$. This protocol is in fact an *all-or-nothing* protocol

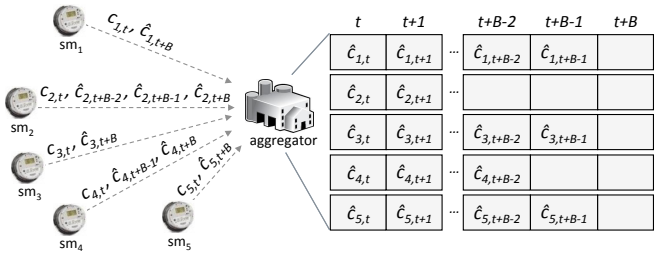


Fig. 1: Illustration of future ciphertext buffering.

since if the aggregator fails to receive one or more ciphertexts from the meters, it cannot obtain any useful information because the sum of $r_{i,t}$ will not be zero.

C. Secure aggregation with fault tolerance

In this section, we present a secure *proactive* aggregation protocol with fault tolerance, using a novel design of *future ciphertext buffering*.

Future ciphertext buffering. Each smart meter i sends two kinds of ciphertexts to the aggregator, namely *current ciphertexts* and *future ciphertexts*. The current ciphertext is given by

$$c_{i,t} = \tilde{x}_{i,t} + r_{i,t} = x_{i,t} + \hat{G}_{i,t}(N, \lambda) + r_{i,t},$$

where $\hat{G}_{i,t}(N, \lambda)$ denotes $G_{i,t}(N, \lambda) - G'_{i,t}(N, \lambda)$. The future ciphertext is given by

$$\hat{c}_{i,t} = \hat{G}_{i,t}(N, \lambda) + r_{i,t} + Lap_{i,t}(\lambda).$$

Note that $Lap_{i,t}(\lambda)$ is a Laplace noise generated solely by i. We assume that the aggregator has a memory buffer so that it can store B future ciphertexts per meter. In a time slot, each meter i sends one current ciphertext and b future ciphertexts ($0 < b \leq B$). The purpose is for i to always try to fill the aggregator's buffer with its future ciphertexts, as shown in Fig. 1. Suppose, for example, that in time slot t, the aggregator already has $B - 1$ future ciphertexts: $\hat{c}_{i,t}, \hat{c}_{i,t+1}, \hat{c}_{i,t+2}, \ldots \hat{c}_{i,t+B-1}$. Then, i sends just two ciphertexts in t: one current ciphertext $c_{i,t}$ and one future ciphertext $\hat{c}_{i,t+B}$. If the aggregator successfully receives the ciphertexts, it stores $\hat{c}_{i,t+B}$ in its buffer. If i fails in time slot t, so it does not report in that round, then in the next time slot $t + 1$, it will try to send one current ciphertext $c_{i,t+1}$ and two future ciphertexts $\hat{c}_{i,t+B}$ and $\hat{c}_{i,t+1+B}$. In steady-state normal operation (i.e., without failures), the protocol sends one current ciphertext and one future ciphertext per time slot. Hence, although it adds redundancy for proactive recovery, it is in fact more bandwidth efficient than the reactive protocol in [1].

Note that current ciphertexts and future ciphertexts contain two random numbers in common: $\hat{G}_{i,t}(N, \lambda)$ and $r_{i,t}$. Assume that in time slot $t - B$, i sends a future ciphertext $\hat{c}_{i,t}$. After B time slots, i will send a current ciphertext $c_{i,t}$. If a current ciphertext and a future ciphertext contain the same t, $\hat{G}_{i,t}(N, \lambda)$ in $c_{i,t}$ is equal to $\hat{G}_{i,t}(N, \lambda)$ in $\hat{c}_{i,t}$ and $r_{i,t}$ in $c_{i,t}$ is equal to $r_{i,t}$ in $\hat{c}_{i,t}$.

Buffer size requirements. Assume that electricity consumption measured by a smart meter is a 32-bit value, and the number of meters is 2^{20}. Then, the modulus value for modular addition is 2^{52}, which means that the size of a future ciphertext is 52bits. Therefore, if we assume that B is 2^{10}, the total buffer size required for the future ciphertexts is $52 \times 2^{20} \times 2^{10}$ bits which is approximately equal to 6.5GB. By the standard of today's consumer device technologies, it is a low memory requirement (for reference, portable terabyte disk drives sell at about US$50).

Decryption. In the case that the aggregator receives the current ciphertexts from all the meters, it can decrypt the sum using only the current ciphertexts:

$$\sum_{i=1}^{N} c_{i,t} = \sum_{i=1}^{N} \tilde{x}_{i,t} + \sum_{i=1}^{N} r_{i,t} = \sum_{i=1}^{N} x_{i,t} + Lap_t(\lambda).$$

If the aggregator fails to receive a current ciphertext $c_{j,t}$ from meter j in time slot t, it can use j's future ciphertext $\hat{c}_{j,t}$ buffered to decrypt the sum. Particularly, if only one smart meter j failed and did not report, the aggregator gets the sum as follows:

$$\begin{aligned}
\sum_{i=1,i\neq j}^{N} c_{i,t} + \hat{c}_{j,t} &= \left(\sum_{i=1,i\neq j}^{N} \tilde{x}_{i,t} + \sum_{i=1,i\neq j}^{N} r_{i,t} \right) \\
&\quad + \left(\hat{G}_{j,t}(N, \lambda) + r_{j,t} + Lap_{j,t}(\lambda) \right) \\
&= \sum_{i=1,i\neq j}^{N} x_{i,t} + \sum_{i=1,i\neq j}^{N} \hat{G}_{i,t}(N, \lambda) + \hat{G}_{j,t}(N, \lambda) \\
&\quad + \sum_{i=1,i\neq j}^{N} r_{i,t} + r_{j,t} + Lap_{j,t}(\lambda) \\
&= \sum_{i=1,i\neq j}^{N} x_{i,t} + \sum_{i=1}^{N} \hat{G}_{i,t}(N, \lambda) + \sum_{i=1}^{N} r_{i,t} + Lap_{j,t}(\lambda) \\
&= \sum_{i=1,i\neq j}^{N} x_{i,t} + Lap_t(\lambda) + Lap_{j,t}(\lambda).
\end{aligned}$$

The aggregator will obtain the sum not including $x_{j,t}$, and the accuracy degrades by $Lap_{j,t}(\lambda)$ as a result. In general, the aggregator is still able to calculate the sum when more meters fail, albeit at the cost of a larger error. Specifically, if w smart meters failed, $w + 1$ values of Laplace noise will remain. Therefore, the proposed scheme generates $O(\sqrt{w+1})$ error when w smart meters failed to report.

Join of new meter. Assume that a smart meter, say i, joins just before time slot t. i will have to send its shared keys to k meters it chooses to partner with and upload B future ciphertexts to the aggregator: $\hat{c}_{i,t}, \hat{c}_{i,t+1}, \hat{c}_{i,t+2}, \ldots \hat{c}_{i,t+B-1}$, before it sends the first current ciphertext $c_{i,t}$ in time slot t. Let m be a chosen partner of i's who accepts a new shared key from i. m will also need to upload new ciphertexts $\hat{c}_{m,t}, \hat{c}_{m,t+1}, \hat{c}_{m,t+2}, \ldots \hat{c}_{m,t+B-1}$ to the aggregator. Thus, the communication cost of supporting meter join is $O(k \times B)$.

Leave of existing meter. Suppose that i decides to leave the network with effect in time slot t. In time slot $t-B$, i will have to inform the aggregator of its leave decision in advance, and the aggregator will broadcast a leave announcement message containing the id of i and t, denoted by *leave(i,t)*.

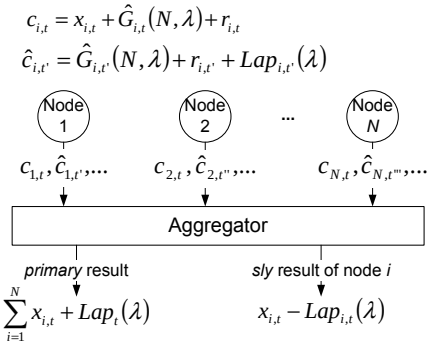


Fig. 2: Our fault-tolerant secure aggregation protocol. At time t, smart meters send both current ciphertext ($c_{i,t}$) and future ciphertexts ($\hat{c}_{i,t'}$).

The leave decision of i will affect two types of meters: those chosen by i to be partners, and those who chose i to be a partner. Let m be a partner of i's (whether m chose i or was chosen by i). It shares a key $sk_{i,m}$ with i. Before time slot t, m will still use the shared key for its current ciphertext and any future ciphertexts before t, but it will discard the key in t and stop using it for future ciphertexts starting from that time slot.

Let l be a smart meter which chose i as a partner, and hence shares key $sk_{l,i}$ with i. When l receives the leave notification about i, it will need to find another partner to replace i starting from time slot t and maintain the number of meters it chooses to partner with at k. Since there are at most $k + C$ smart meters, where C is a constant and $k \gg C$, which chose i as a partner, at most $k + C$ meters will need to find another partner. Therefore, the total communication cost for meter leave is $O(1 + k + C) = O(k)$.

Note that a meter, say i, that failed will leave the system abruptly, without executing the proper procedure. In this case, the aggregator detects the failure after not hearing from i for some time duration. It then initiates handling of the abrupt leave, by broadcasting a leave message on behalf of i with t set to the time slot of the last future ciphertext the aggregator has from i.

D. Privacy analysis of proposed secure aggregation protocol

Theorem 1. *Our secure aggregation protocol with fault tolerance, which is shown in Fig. 2, provides 2ϵ-DP if $\lambda = GS_f/\epsilon$.*

Proof: To analyze the privacy of the protocol, the main task is to account for the information available in the future ciphertexts. To do so, suppose that based on the information it receives in the protocol, the aggregator performs $N+1$ queries on the data. The first query is on the entire dataset about all the meters. Let Z denote the set of smart meters that have failed. If Z is empty, the query result is $\sum_{i=1}^{N} x_{i,t} + Lap_t(\lambda)$, which satisfies ϵ-differential privacy. Otherwise, the aggregator obtains $\sum_{i=1, i\notin Z}^{N} x_{i,t} + \sum_{z\in Z} Lap_{z,t}(\lambda) + Lap_t(\lambda)$ which satisfies ϵ-differential privacy as well, albeit with a larger error than in the case of no failures. We refer to the result of the first query as the *primary result*.

The other N queries are run on disjoint datasets, specifically one for each of the N smart meters. A curious aggregator can abuse meter i's future ciphertext $\hat{c}_{i,t}$ in time slot t to infer $x_{i,t}$, by computing

$$\begin{aligned} c_{i,t} - \hat{c}_{i,t} &= \left(x_{i,t} + \hat{G}_{i,t}(N,\lambda) + r_{i,t}\right) \\ &\quad - \left(\hat{G}_{i,t}(N,\lambda) + r_{i,t} + Lap_{i,t}(\lambda)\right) \\ &= x_{i,t} - Lap_{i,t}(\lambda), \end{aligned}$$

even if it successfully received $c_{i,t}$ and should not have to use $\hat{c}_{i,t}$. By doing this, for each i, the aggregator can calculate i's $x_{i,t}$ perturbed by $Lap_{i,t}(\lambda)$, i.e., $x_{i,t} - Lap_{i,t}(\lambda)$. Since the Laplace distribution has a symmetric shape about the mean zero, $x_{i,t} - Lap_{i,t}(\lambda)$ also provides ϵ-DP on data $x_{i,t}$. We refer to the results of the N queries as the *sly results*. The $x_{i,t}$'s on which the N queries are performed are disjoint datasets, Hence, ϵ-differential privacy is achieved by the parallel composition rule ([12, Theorem 4]). In time slot t, from the perspective of meter i, the two query results which contain $x_{i,t}$ satisfy ϵ-DP. As a result, by the sequential composition rule ([12, Theorem 3]), 2ϵ-DP is satisfied. ■

E. Error optimization by meter failure probability

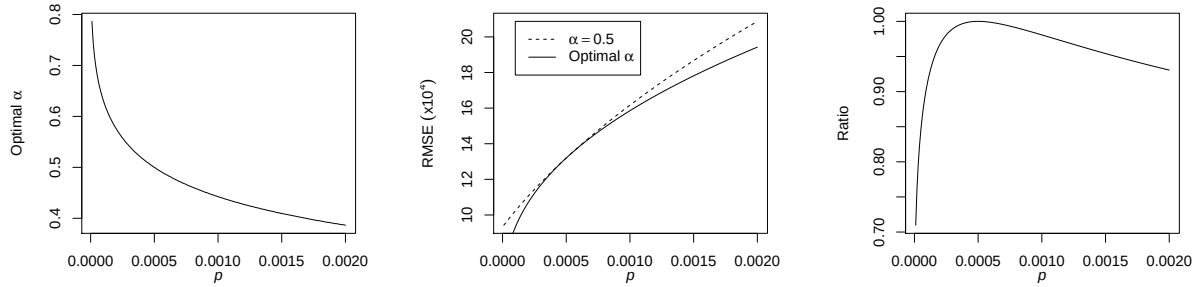
The privacy budget of the proposed protocol is a sum of two parts, one given to the primary result and the other given to the sly results (see proof of Theorem 1). Assume that in each time slot, meters fail independently with probability p. In this section, we describe how to reduce the errors of query results by optimizing the division of the DP budget given p. For example, in the proof of Theorem 1, we divided the privacy budget ϵ equally between the primary result and the sly results, and achieved 2ϵ-DP overall. However, we can achieve the same 2ϵ-DP by a different division of the budget, for example, 1.5ϵ for the primary result and 0.5ϵ for the sly results. In general, for our fault-tolerant aggregation protocol to support ϵ-DP, the s.d. of noise needed is

$$\sqrt{2\left(\frac{GS_f}{\alpha}\right)^2 + 2\cdot N\cdot p\cdot\left(\frac{GS_f}{\epsilon-\alpha}\right)^2}, \quad (\epsilon > \alpha > 0), \quad (1)$$

where $N \cdot p$ is the expected number of smart meters that failed to transmit. α is the privacy share of the primary result and $\epsilon - \alpha$ is that of the sly results, so that ϵ-DP is achieved overall. Note that the s.d. quantity (1) is in fact equal to the root mean square error (RMSE) of the per-period aggregation results. As such, it quantifies the overall error of the aggregation protocol.

The first term inside the square root of (1) is the variance of noise generated in a distributed manner to perturb the primary result, which depends on α. The second term is the variance of noise introduced by the future ciphertexts of individual smart meters that failed, which depends on p and α. Intuitively, if p is small, most of the time the aggregator successfully receives the current ciphertexts from all the meters, so that it can calculate the sum without any future ciphertexts. Hence, if we set α larger approaching ϵ, the s.d. of noise becomes smaller. On the other hand, if p is high, α should be smaller to reduce the impact of the second term. The optimal α for minimizing the s.d., which is the aggregation error, can be calculated given p.

Fig. 3 illustrates analytical results for the aggregation error when ϵ is 1, GS_f is 33,000W, and N is 2,000. Fig. 3 (a) shows



(a) Optimal α minimizing the RMSE, as a function of failure probability p.

(b) Comparison of RMSE between the use of (i) fixed α, (ii) optimal α by p.

(c) Ratio of RMSEs: Optimal α approach to fixed α approach.

Fig. 3: Minimizing the RMSE by dividing the overall privacy budget between the primary result and the sly results, when supporting ϵ-DP.

the optimal α values that minimize the error with respect to p. The optimal α decreases from 0.787 to 0.386 as the failure probability increases from 0.0001 to 0.002. Fig. 3 (b) compares the aggregation errors for different p when α is fixed at 0.5, versus when the optimal α values are applied. As shown in Fig. 3 (c), by using the optimal α, the aggregation error is reduced by 29% when p is 0.0001 and 7% when p is 0.002.

F. Error reduction by individual sensitivity

As shown in (1), when p is large, the noise for protecting the sly results dominates that for the primary result. In this section, we show that the noise for protecting the sly results can be further reduced using a notion of *individual sensitivity*.

Definition 2. (Individual sensitivity) *The **individual sensitivity** of a smart meter, say i, is the quantity*

$$S_i = \max_{-\infty<t<\infty} x_{i,t},$$

where $x_{i,t}$ is the consumption data measured by i in time slot t.

Hence, if i measures the electricity consumption of a household, S_i is the maximum amount that the household can consume in a time period. It is related to GS_f by

$$GS_f = \max_{1\le i\le N} S_i. \tag{2}$$

In the previous section, in order to perturb the sly results, every smart meter adopts the same system parameter, GS_f, to generate noise in the future ciphertexts. However, we can reduce the noise by replacing the GS_f in future ciphertexts by S_i. The s.d. of noise is then given by

$$\sqrt{2\left(\frac{GS_f}{\alpha}\right)^2 + 2\cdot p\cdot\sum_{i=1}^{N}\left(\frac{S_i}{\epsilon-\alpha}\right)^2}, \quad (\epsilon>\alpha>0). \tag{3}$$

It is obvious that (3) is less than or equal to (1) due to (2).

Theorem 2. *Our fault-tolerant secure aggregation protocol based on individual sensitivity, shown in Fig. 4, supports ϵ-DP, if $\lambda_1 = GS_f/\alpha$ and $\lambda_2 = S_i/(\epsilon-\alpha)$, where $\epsilon>\alpha>0$.*

Proof: In time slot t, the aggregator obtains the primary result and N sly results. Since the primary result is perturbed by Laplace noise whose noise scale is λ_1, α-DP is satisfied. Since the N sly results obtained are parallel in terms of privacy,

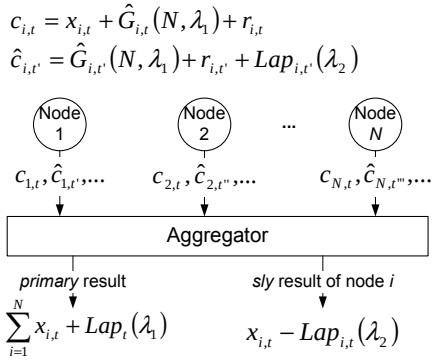


Fig. 4: Our fault-tolerant secure aggregation protocol based on individual sensitivity in future ciphertexts $\hat{c}_{i,t'}$.

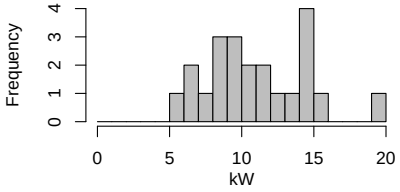


Fig. 5: Histogram of maximum power consumed, as recorded for 22 households in the U.K.

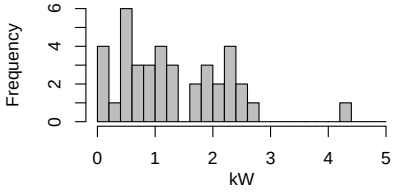


Fig. 6: Histogram of maximum power consumed, as recorded for 39 electricity branches at a research center in Singapore.

and they are perturbed by Laplace noise whose noise scale is λ_2, each sly result guarantees $(\epsilon-\alpha)$-DP ([12, Theorem 4]). Consequently, by the sequential composition rule ([12, Theorem 3]), ϵ-DP is assured. ■

Fig. 5 and Fig. 6 illustrate the reduction in error made possible by individual sensitivity based on real-world traces of electricity consumption. Fig. 5 shows a histogram of the maximum power[1] consumed by 22 households in the U.K.

[1]Since the reporting time period is fixed, power is directly proportional to electricity by a fixed scaling factor.

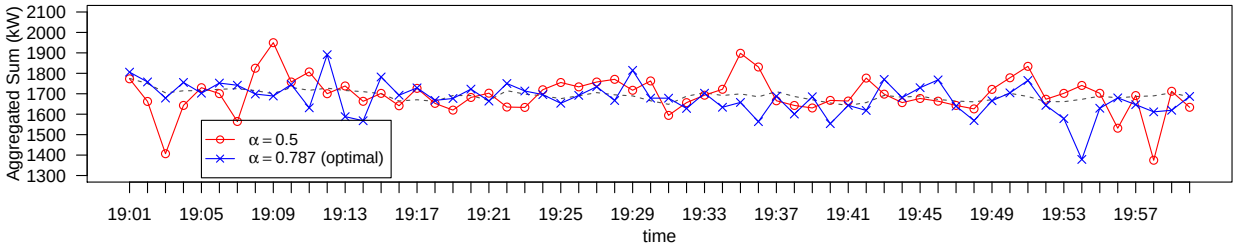


Fig. 7: The computed aggregated sum of power loads for different α as a function of time, when $p = 0.00001$, $GS_f = 33$kW, and $\epsilon = 1$. The optimal α is 0.787.

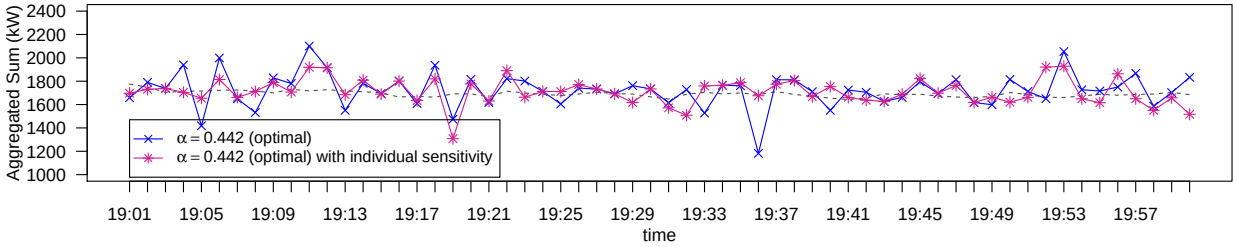


Fig. 8: The computed aggregated sum of power loads using global sensitivity vs. individual sensitivity as a function of time, when $p = 0.001$, $GS_f = 33$kW, and $\epsilon = 1$. The optimal α is 0.442.

from January 2008 to December 2009 [16]. In the histogram, the largest value is 19,681W and the sum of all the maximum powers is 242,015W. If we take the largest of the maximum powers as the global sensitivity, and each maximum power itself as the individual sensitivity, then if the expected number of failed meters is one (i.e., $p = 1/22$) and $\alpha = 0.5$, we can obtain the ratio of (3) to (1) as:

$$\sqrt{\frac{2 \cdot (GS_f/\alpha)^2 + 2 \cdot p \cdot \sum_{i=1}^{N} \left(S_i/(\epsilon - \alpha)\right)^2}{2 \cdot (GS_f/\alpha)^2 + 2 \cdot p \cdot N \cdot \left(GS_f/(\epsilon - \alpha)\right)^2}} \approx \sqrt{0.67} \approx 0.82$$

Therefore, we can expect an 18% reduction in the aggregation error, by using individual sensitivity instead of global sensitivity.

Fig. 6 shows a histogram of the maximum powers we recorded for 39 branches in the electricity network of the Advanced Digital Sciences Center (ADSC), Singapore, from May 28, 2012 to May 7, 2013. The measurement interval is roughly two seconds and the largest measured value is 4,387W. In this setting, we can expect a 26% reduction in the aggregation error, which is more than the 18% in the U.K. household scenario. This is because at ADSC, one particular branch of the electricity network consumed a lot more than the other branches. In practice, it can be difficult for i to estimate its own S_i accurately beforehand. However, it is feasible to exploit individual sensitivity by grouping smart meters by, for example, the types of housing units (e.g., square footage, construction technology, etc, of residential flats) they belong to, and assign an individual sensitivity to each type.

V. Evaluation

To evaluate our aggregation protocol, we make use of a state-of-the-art generator for electricity traces [17]. The generator produces synthetic electricity consumption for one household at one minute resolution, based on realistic appliance and lighting models, over one specific day of the year (so that typical seasonal activities and weather conditions can be considered, for example). Based on it, we produce traces for 2,000 households. We specify the number of residents in each household using U.K. statistics on household sizes in 2011 [13]. We select the day to be a weekday in January. For the appliances in a household, we choose them randomly among 33 available ones. For differential privacy, ϵ is set to 1 and the global sensitivity is set to 33kW, which is the sum of power demands of all the appliances and lights. Each household also records its own individual sensitivity, which is the sum of power demands of its own appliances and lights.

Fig. 7 and Fig. 8 illustrate a trace of the actual total consumption and the noisy total consumption for DP, from 7pm to 8pm. The trace of actual sums is shown as the dashed line in both figures. We run two separate simulations, corresponding to a low communication failure scenario ($p = 0.00001$) and a high failure scenario ($p = 0.001$), respectively[2]. Fig. 7 shows the traces when p is 0.00001, and compares the two cases when $\alpha = 0.5$ and when $\alpha = 0.787$, respectively. The latter α value is the optimal α that minimizes the error according to our analysis (Section IV-E). We measure the closeness between the sequences of actual and noisy sums by the RMSE, i.e.,

$$\text{RMSE} = \sqrt{\frac{1}{T} \cdot \sum_{t=1}^{T} (\hat{r}_t - r_t)^2},$$

where $T = 1,440$ is the number of time slots in one day, $\hat{r}$ is the noisy sum, and r is the true sum. Fig. 9 shows the RMSE of the proposed aggregation protocol under different settings. For $p = 0.00001$, the RMSE is 65,928W when $\alpha = 0.5$, whereas it is 46,166W when α is optimal (i.e., $\alpha^* = 0.787$). The optimal α provides a 30% reduction in RMSE compared with the fixed α.

[2]As meters are sampled every minute, $p = 0.00001$ means a meter fails once every 69.4 days on average, while $p = 0.001$ means a meter fails once every 16.7 hours on average.

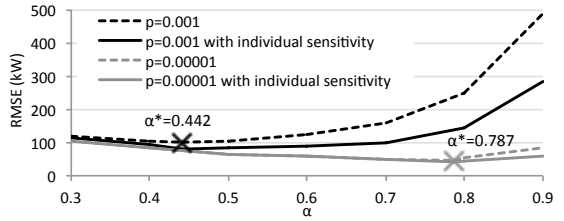


Fig. 9: Comparison of RMSE under different settings of the proposed aggregation protocol. A cross indicates the α^* (optimal α) computed by our optimization algorithm for the corresponding p.

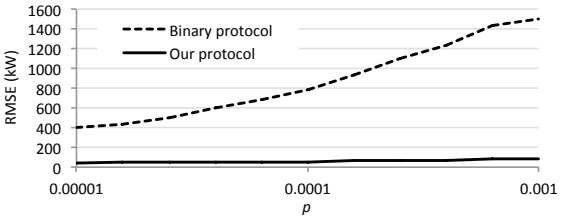


Fig. 10: Comparison of RMSE between the proposed aggregation protocol and the binary protocol [3].

Fig. 8 shows two traces when the optimal α (0.442) is used with global sensitivity and individual sensitivity, respectively. p is set to 0.001 in both cases. Because of more communication failures in this setting, the DP-induced noise is generally higher than in the case of $p = 0.00001$ in Fig. 7. As shown in Fig. 9, for $p = 0.001$ and $\alpha = 0.442$, the RMSE is 100,895W under global sensitivity only, and 80,222W when individual sensitivity is applied. Individual sensitivity contributes to an approximately 20% reduction in the RMSE.

Fig. 10 compares the RMSE of our protocol under optimal α and individual sensitivity, with a state-of-the-art proactive binary protocol [3] whose fault tolerance is the same as our protocol's. The figure shows that the binary protocol generates 9.6 times and 18.7 times larger RMSE than our protocol when p is 0.00001 and 0.001, respectively. Our significantly improved accuracy is mainly because the amount of noise needed for privacy in our protocol does not depend on N, whereas that of the binary protocol does (see Table I).

VI. Conclusion

We have presented a fault-tolerant aggregation protocol for smart meters to report consumption to an untrusted aggregator with assured differential privacy. Our fault-tolerance approach is proactive and based on a novel design of future ciphertexts. We proved the differential privacy of our protocol. We also analyzed its communication complexity in normal operation and when meters join or leave. Although the proposed protocol distributes trust and noise for differential privacy among the meters, we showed that it can gracefully tolerate missing reports due to communication failures. Computational efficiency at the meters is assured by modular addition-based encryption.

Compared with a state-of-the-art proactive binary protocol [3], our protocol has the same fault tolerance but it is much more bandwidth-efficient. Its aggregation error does not grow with the total number of meters, unlike the binary protocol. We presented minimization of the aggregation error based on the communication failure probability and a notion of individual sensitivity. Simulations driven by realistic energy traces showed that the error reduction in practice is significant. Compared with a state-of-the-art reactive protocol [1], our protocol is bandwidth efficient. More importantly, the reactive protocol will break when communication failures prevent encryption keys to cancel out in a three-message exchange process. These failures are quite possible in practical operation, but they will not affect the fault tolerance of our protocol.

Acknowledgment

This research was supported in part by Singapore Agency for Science, Technology, and Research (A*STAR) under the Human Sixth Sense Programme, in part by U.S. NSF under grant numbers CNS-0963715 and CNS-0964086, in part by Singapore MOE under SRG-ISTD-2013-060 and an SUTD-ZJU grant award, and in part by China NSFC under 61028007.

References

[1] G. Ács and C. Castelluccia. I have a dream!: differentially private smart metering. In *Proc. of Information Hiding Conference*, May 2011.
[2] S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering, and demand response in electricity markets. Technical Report CSEMWP105, Center for the Study of Energy Markets, October 2002.
[3] T.-H. Chan, E. Shi, and D. Song. Privacy-preserving stream aggregation with fault tolerance. In A. Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 200–214. Springer Berlin Heidelberg, 2012.
[4] C. Dwork. Differential privacy. In *Proc. of ICALP*, 2006.
[5] C. Dwork. Differential privacy: A survey of results. In *Proc. of TAMC*, April 2008.
[6] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: privacy via distributed noise generation. In *Proc. of EUROCRYPT*, May–June 2006.
[7] G. W. Hart. Residential energy monitoring and computerized surveillance via utility power flows. *IEEE Technology and Society Magazine*, June 1989.
[8] W. Heck. Smart energy meter will not be compulsory. http://vorige.nrc.nl/international/article2207260.ece/Smart_energy_meter_will_not_be_compulsory.
[9] S. Kotz, T. J. Kozubowski, and K. Podgórski. *The Laplace Distribution and Generalizations*, pages 46–47. Birkhäuser, 2001.
[10] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker. Inferring personal information from demand-response systems. *IEEE Security and Privacy*, 8(1), Jan.–Feb. 2010.
[11] F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. In *Proc. of IEEE FOCS*, October 2007.
[12] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proc. of ACM SIGMOD*, June–July 2009.
[13] Office for National Statistics. Families and households, 2001 to 2011. http://www.ons.gov.uk/ons/rel/family-demography/families-and-households/2011/rft-tables-1-to-8.xls.
[14] E. L. Quinn. Smart metering and privacy: Existing law and competing policies. *A report for the Colorado Public Utilities Commission*, 2009.
[15] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proc. of ACM SIGMOD*, June 2010.
[16] I. Richardson and M. Thomson. One-minute resolution domestic electricity use data, 2008-2009 [computer file]. colchester, essex: Uk data archive [distributor], october 2010. sn: 6583, http://dx.doi.org/10.5255/ukda-sn-6583-1. http://discover.ukdataservice.ac.uk/catalogue?sn=6583.
[17] I. Richardson, M. Thomson, D. Infield, and C. Clifford. Domestic electricity use: A high-resolution energy demand model. *Energy and Buildings*, 42(10):1878 – 1887, 2010.
[18] E. Shi, R. Chow, T. h. Hubert Chan, D. Song, and E. Rieffel. Privacy-preserving aggregation of time-series data. In *Proc. of NDSS*, 2011.
[19] G. P. Zachary. Saving smart meters from a backlash. *IEEE Spectrum*, August 2011.