Di-PriDA: A Privacy-preserving Meter Querying System for Smart Grid Load Balancing

Xiaojing Liao†, David Formby†, Carson Day‡, Raheem A. Beyah†
†Communications Assurance and Performance (CAP) group
‡National Electric Energy Testing Research and Applications Center (NEETRAC)
School of Electrical and Computer Engineering, Georgia Institute of Technology

***Abstract*—The smart grid will utilize appliance-level control to provide sustainable power usage and flexible energy utilization. Given the privacy and efficiency concerns of the smart grid system, we propose a cost-efficient platform called Di-PriDA for support of appliance-level peak-time load balance control in the smart grid, in which data analysis operations are achieved in a privacy-preserving manner utilizing distributed differential privacy. Additionally, Di-PriDa ensures rigorous provable privacy and guarantees the result accuracy. We validate the efficiency and accuracy of the proposed scheme under a real-world power usage dataset.**

I. Introduction

The future electrical grid, i.e., smart grid as shown in Figure 1, utilizes appliance-level load control policy to provide sustainable power usage and flexible energy utilization. For example, to respond to a rapid power consumption increase in peak times among neighborhoods, the *peak-time load balancing control* for smart grid can temporarily (to allow time to start up a larger generator) or continuously (in the case of limited resources) shut down the appliances which are not in use but connected to the circuit. Due to privacy concerns [1], most non-invasive techniques proposed in the literature use battery-based load hiding (BLH) to hide the appliance readings, which, however, does not support the data analysis at the controller. In other systems, cryptographic methods such as homomorphic encryption (HE) are used to enable privacy-preserving data analysis. However, as an invasive cryptographic method, its development requires infeasible change to the existing smart grid. Also, the homomorphic encryption process is not regarded as efficient for resource-constrained devices in the smart grid system. A comparison of our technique with others is illustrated in Table I.

In this ongoing work, we propose a novel privacy-preserving load data analysis platform, Di-PriDA, based on distributed differential privacy (DP) [2], which is non-invasive and efficient. The contributions of our proposed work are as follows: First, we explore the distributed top-k differential privacy problem to propose a privacy-preserving load analysis mechanism for appliance-level peak-time load balance control. Second, we show the provable privacy and the upper bound of the error rate for our scheme theoretically. Our experiments, based on a real-world dataset, indicate the efficiency and validity of our scheme.

II. Di-PriDA: System for Peak-Time Load Balance

Our proposed scheme has three steps. First, when the concentrator obtains the query request Q_t from the control center, the concentrator fuzzes the parameters of the query

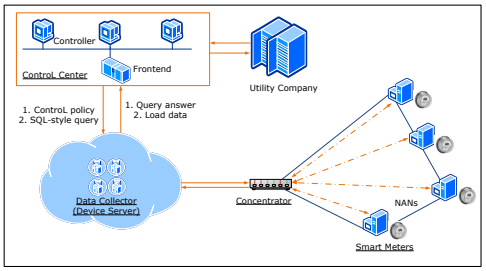


Fig. 1. Overview of the NAN based smart grid system.

TABLE I. A comparison of Related Works with our scheme.

	Cryptographic Approach			Non-cryptographic Approach		
	Our work	LLL10	RV13	RSM11	MMA11	YLQ12
Category	DP	HE	HE	BLH	BLH	BLH
High-Efficiency	✓	✗	✗	✓	✓	✓
Provable Security	✓	✗	✓	✗	✗	✗
Fine-grained control	✓	✓	✓	✗	✗	✗
Non-invasive	✓	✗	✗	✓	✓	✓

request, and then distributes the new query request Q'_t to the smart meters of each house. Second, the smart meter of each house i answers the query Q'_t, and then encrypts the query answer $QA = \{< a_i, t_i, p_{a_i} >\}$ with the controller's public key K_p, where a_i, p_{a_i}, t_i are the UID of the appliance, the power consumption and the timestamp respectively. With the encrypted query answers from each smart meter through the secure channel, provided by TLS, the concentrator adds noise into the set of the encrypted query answers, and then returns k answers among them by uniformly sampling the set. Finally, the controller decrypts the query answers with its private key K_s. The details of the scheme are shown in Algorithm 1.

In *Query Initialization*, the fuzzy parameter m for query distribution is computed as $m = \frac{r_s \ln(e+\varepsilon^2 kOPT)}{\varepsilon}$, where r_s is the sample rate of the data collector, k is the number of the returned query answers, ε is the privacy budget and OPT is the sum of real top-k appliances power usage. In *Query Response*, n_i is the power consumption noise, $n_i = LAP(\frac{\Delta f_s}{\varepsilon})$, $\Delta f_s < 1$. In *Response Process*, the noise the concentrator added is computed as $c_i = ne^{\varepsilon f_i} - f_i$, $c = \sum c_i$, where n is the size of query answers set, ε is the privacy budget for differential privacy and f_i is the frequency of the appliance appears, i.e., $f_i = \frac{\#\ of\ the\ pattern\ `H_{K_p}(i||a_i)'}{n}$. Di-PriDA is implemented as privacy modules to be plugged in the smart meter and the concentrator.

III. Security and Performance Analysis

Considering the protection from the untrusted controller, during the query, only the encrypted query results will be

Algorithm 1 The sequential scheme of algorithms.

Input: Query request q, Privacy budget ε, Meter reading R.

1: **Query Initialization**: The concentrator relaxes the original query Q_t's time range from $[t_{pl}, t_{ph}]$ to $[t_{pl} - m, t_{ph}]$, then forwards the new query Q'_t to each of the smart meters.

2: **Query Response**: After receiving the query request, each smart meter of a house searches the metering reading log R to obtain the answers QA, then adds the noise n_i in the power consumption to generate a fuzzy query answer QA', i.e., $p_{a_i} + n_i$. Finally, the data is encrypted then sent back to to the concentrator.

3: **Response Process**: The concentrator processes the query responses from the smart meters of each house under the following two policies: (1) add c noise query answers based on the frequency f_i of the appliance appears (i.e., $H_{K_p}(i||a_i)$). (2) uniformly sample k distinct items from the set of the query responses including the noise query answers.

4: **Answer Response**: The concentrator returns the k distinct encrypted items to the controller through the data collector. Then, the controller decrypts the message using its private key K_s, and obtains the appliances under idle mode which have the top-k largest power consumptions in peak time.

submitted to the server. In other words, the server obtains nothing but the query result. With the configurable privacy budget ε, the scheme is 3ε-differential privacy. Namely, the removal or addition of a single user's data does not substantially affect the result, thus there is no risk for users to join and answer the query. Also, an honest-but-curious user is not able to obtain the power load of others because they do not communicate with each other directly. Additionally, the communication messages between the user and the controller are protected from eavesdropping and modification by other users, because of the secure communication channel. The untrusted concentrator is not able to obtain the power load of the users because all the power loads along with the user's information were encrypted and blinded to the concentrator.

Theorem 1: The scheme gives 3ε-differential privacy.

Proof: In *Query Response*, as the power consumption noise n_i is added as the Laplace noise, i.e., $n_i = LAP(\frac{\Delta f_s}{\varepsilon})$, the algorithm achieves ε-differential privacy. In *Response Process* (a.k.a., $IA()$), considering $f_c(a_i) < 1$, where $f_c(a_i)$ is the chosen frequency of the appliance a_i, the sensitivity of the chosen frequency $\Delta f_c(a_i) < 1$. With the noises $c(a_i) = ne^{\varepsilon f_c(a_i)} - f_c(a_i)$ added for the appliance a_i and uniformly sampled, the sampled probability of the appliance a_i is $e^{\varepsilon f_c(a_i)}$. For two data sets D_1 and D_2 differing on at most one row, $Pr(IA(D_1)) \leq e^{2\varepsilon} Pr(IA(D_2))$. Hence, the algorithm *InitAnswer()* is 2ε-differential privacy. By the use of Composisity Theorem [2], the scheme we proposed gives 3ε-differential privacy. ■

Considering the accuracy of the scheme, the error rate is defined as $d = \frac{OPT - \sum_{i=1}^{k} p(a_i)}{OPT}$, where OPT is the real sum of the top-k appliances' power usage, and k is the number of the query results.

Theorem 2: The scheme we proposed has the upper bound of the error rate as $\frac{3\ln(e+\varepsilon^2 kOPT)}{\varepsilon OPT}$, where OPT is the real sum of top-k appliances' power usage.

Proof: Assume $S_{2t} : \{a_i : A(a_i) > OPT - 2t\}$, where $A()$ is the sequential scheme we proposed. Hence,

$$A(S_{2t}) \leq \frac{A(S_{2t})}{A(S_t)} \leq \frac{e^{-\varepsilon t}}{\mu(S_t)} \quad (1)$$

Considering the expected results $E[A(a_i)] = (OPT - 2t)(1 - A(S_{2t}))$. As $m = \frac{r_s \ln(e+\varepsilon^2 kOPT)}{\varepsilon} > \frac{r_s \ln \frac{OPT}{t\mu(S_t)}}{\varepsilon}$,

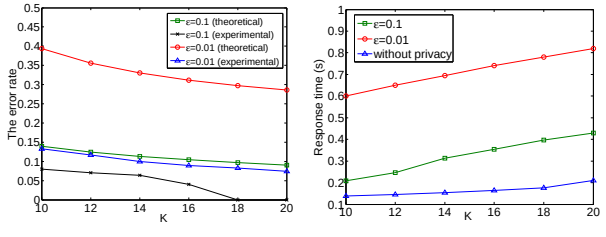
$$1 - A(S_{2t}) > 1 - \frac{m}{r_s OPT} \quad (2)$$

$$E[A(a_i)] \geq OPT - \frac{3\ln(e + \varepsilon^2 kOPT)}{\varepsilon} \quad (3)$$

Therefore, the upper bound of the error rate is $\frac{3\ln(e+\varepsilon^2 kOPT)}{\varepsilon OPT}$. ■

IV. Initial Evaluation

We evaluated the accuracy and the response time of our schemes on a real-world dataset: UMASS SMART* dataset [3]. The simulation is implemented in Python on a PC which had two 3.10 GHz Intel Core i5-2400 processors running the Linux 3.5 kernel. We used pycrypto (a.k.a., Python Cryptography Toolkit) to implement the RSA-OAEP and SHA-2 as instances of the public-key encryption and hash function, respectively. Figure $2(a)$ shows the error rate of the scheme with different privacy budgets ε. Both the theoretical upper bound of the error rate and the error rate in the experiment are presented. Overall, the error rate of the scheme decreases as the number of query results k increases. With the larger privacy budget ε, i.e., $\varepsilon = 0.1$, both the upper bound of the error rate and the experimental error rate are smaller than those with smaller privacy budget, i.e., $\varepsilon = 0.01$. Compared to the upper bound of the error rate under the same privacy budget, the experimental error rate is much lower than the theoretical one. Moreover, when the privacy budget is small, the difference between the upper bound of the error rate and the experimental error rate becomes larger. Overall, the observed error rates based on the experiments are less than 14% when $\varepsilon = 0.01$ and less than 7% when $\varepsilon = 0.1$. Figure 2(b) presents the response time of our scheme with different privacy budgets. To indicate the performance degradation, the response time of our scheme is compared with that without any security mechanism. As the number of query results k increases, the response time of the scheme increases. Also, the smaller privacy budget introduces a larger performance degradation, i.e., when the privacy budget $\varepsilon = 0.01$, the response time becomes larger than that with smaller privacy budget. In our privacy-preserving scheme with privacy budget $\varepsilon = 0.1$, the increase in the response time is below $0.4s$, which is about 105% of that without any security mechanism.



(a) The error rate of our scheme with different privacy budgets

(b) The response time with different privacy budgets

References

[1] I. Rouf, H. Mustafa, M. Xu, et al. Neighborhood watch: security and privacy analysis of automatic meter reading systems. In Proceedings of the 2012 ACM CCS: 462-473.

[2] C. Dwork. Differential privacy. In Automata, languages and programming. Springer Berlin Heidelberg, 2006: 1-12.

[3] S. Barker, A. Mishra, D. Irwin, et al. Smart*: An open data set and tools for enabling research in sustainable homes. In Proceedings of the 2012 SustKDD.