ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

# A Privacy-Preserving Data-Sharing Framework for Smart Grid

[1] SK.Dayeen Sidhika, [2] Dr.P.Venkateswara Rao, [3] S.Yamini, [4] T.Hemalatha
[2] Professor & HOD
[1][2][3][4] Narayana Engineering College, Gudur

**Abstract: Distributed energy resources (ERs), featured with small-scale power generation technologies and renewable energy sources, are considered as necessary supplements for smart grid. To ensure that merged resources contribute effectively to the grid, data generated by consumer side should be shared among the ERs. However, it also introduces challenges of the protection of consumer privacy. To address these difficulties, we propose a new framework to share data in smart grid by leveraging new advances in homomorphic encryption and proxyre-encryption. Our proposed framework allows ERs to analyze consumer data while ensuring consumer privacy. An additional benefit of our proposed framework is that consumer data is transmitted over the smart grid only once. Furthermore, we present a concrete scheme falling into the proposed framework. Extensive analysis shows that the concrete scheme is secure and efficient. Index Terms—Cloud computing, data sharing, homomorphic encryption, privacy-preserving, proxy re encryption, smart grid.**

## INTRODUCTION

THERE HAVE been several instances when power grids across the globe risked catastrophic failure [1]. Oftentimes, power outages are caused by localized defects in the electricity networks. If a small defect is not dealt in a proper and timely manner, it could lead to a cascading failure of the power supply network. For example, a power outage on the east coast of the U.S. and Canada in 2003 was such a case. A power line was damaged by a tree in Cleveland, OH, USA. Making matters worse is that nearby lines became overloaded and overheated by rerouted power and sagged from the excessive heat. This eventually tripped circuit breakers after these lines contacted trees. Approximately 50 million people in the Northeast U.S. and part of Canada were left without power for several days [2]. Power outages can also be caused by overloaded electrical circuits. Electricity consumption is higher during hotter summer days. In some cases electrical demands may exceed power grid capacity. In such cases appliances should be turned off to conserve energy or additional resources should be added to grid to compensate demands. If left unaddressed, an overloaded power grid could fail, resulting in blackouts. It is thus crucial that we monitor power grid systems in real-time to ensure that abnormalities are dealt with promptly and effectively. Smart grids have recently been gaining popularity. They support real-time diagnosis and can react to avoid failures and blackouts [3]–[7]. The major difference between the smart and traditional power grids is information flows. In the traditional power grid, there exists only one-way electrical flows, i.e., electricity utilities only deliver power to consumers. In contrast, smart grids allow for two-way information flow communications. As shown in Fig. 1, the two-way information flows in the smart grid are almost parallel to that of the oneway power flows. However, a control center is also involved in information flows. A control center collects data with which it can decide on how to alter a grid. With electricity consumption reports a control center can analyze consumers electricity consumption data and forecast electricity consumption, and adjust power generation accordingly over a given period. Regular electricity consumption reports are key in smart grid efficacy. On the other hand, the collected electricity consumption reports should be secured to preserve consumer privacy [7], [8]. To operate reliable and resilient smart grids it is paramount that we address security and privacy concerns through established cryptographic schemes. Traditional encryption schemes are quite suitable for single energy source configurations. In these cases, a single control center, oftencontrolled by a government or government affiliated party can be cou nted on as a trusted confidant. This can become troublesome for more complex arrays where supplemental sources from small-scale generation and renewable energy projects come into play [9], [10]. Such grids are referred to as “microgrids.” Not all the distributed energy resources (ERs) are under direct government control. Therefore, trusting these entities is questionable. A dilemma of balancing between consumer privacy and enabling ERs to freely analyze records becomes apparent. A trivial solution is to anonymize data before sending itto ERs for analysis. However, it would significantly increase the communication costs; massive anonymized data needs to be sent to every resource. An alternative is to let some third party perform analysis instead of ERs. Only analysis results are sent to ERs. Nevertheless, the third party would be privy to analysis results. This is undesirable by competing businesses and privacy advocates. To the best of our knowledge, there is no efficient approach so far to this problem in the context of privacy-preserving smart grid. In this paper, we aim to address the above challenge and propose a framework for data sharing in smart grid. The contributions of this paper are twofold. 1) First, we propose a novel data-sharing framework for smart grid, where we combine the two popular infrastructures: a) the smart grid and b) cloud computing. In particular, we allow the electricity consumption reports generated in smart grid to be stored in the cloud. Distributed ERs can obtain the statistics and analysis results from the cloud computing. Hence, our proposed framework can take advantages of cloud computing for the smart grid.

2) Second, our proposed framework makes use of the homomorphic encryption technique to facilitate the statistics and analysis on the encrypted electricity consumption reports, and the proxy re-encryption technique to keep the statistics and analysis results secret from the cloud. A. Differences Between the Conference Version and the Current Version The extended abstract of this paper appeared in [11]. The main difference between the conference version and our current version is our communication model. In the conference version, all the electricity consumption reports are encrypted under the same public key, while they are encrypted under the corresponding public keys in the current version. That is, the framework proposed in the current version is a more generalized one. Due to the change of communication model, we propose a new framework and a concrete scheme. Furthermore, we implement the concrete scheme to show the efficiency. The remainder of this paper is organized as follows. In Section II, we present the system model, security model, and the design goal. Then, we propose the data-sharing framework in Section III, followed by security analysis and performance evaluation in Section IV. We also present related works in Section V. Finally, we draw our conclusions in Section VI.

## II. MODELS AND DESIGN GOAL

In this section, we present the system model, security model, and design goal. A. System Model In this paper, we only focus on how the electricity consumption reports are securely shared among the distributed generation resources. In particular, we take advantages of the data-as-a-service (DaaS) model in cloud computing, where the system is composed of the following parties: the trustedauthority (TA), many electricity consumers (ECs), many ERsand the cloud server as shown in Fig. 2. The TA is responsible for generating the system parameters and the certificate for the public key of each ER. The ECs produce the electricity consumption reports that are outsourced to the cloud server. To achieve the confidentiality, the electricity consumption reports should be encrypted by using the public key of the corresponding ER where the consumed electricity comes from. In order to make a smart decision on the power generation, price and others, each ER would like to do analysis on the electricity consumption reports corresponding to itself or other ERs. Before doing the analysis, the ER should obtain the analysis rights from other ERs. B. Security Model We assume that the cloud server is honest-but-curious as many literatures related to cloud computing [12]–[14]. That is to say, the cloud server will follow the proposed framework faithfully, but could launch passive attacks to get secret information as much as possible. In particular, the cloud server is interested in getting the content of electricity consumption reports or analysis results, but they would not modify the communication data with other entities or collude with other entities. The ERs want to get the analysis results from the cloud server so that they can plan more effectively and efficiently to produce an adequate supply of electricity to serve their local needs. It plays an important role in making our power grid more reliable and resilient since we must balance the grid by matching electricity supply with demand exactly to avoid power grid failure and blackout. Meanwhile, malicious ERs may try to access electricity consumption reports or get the analysis results beyond their analysis rights. C. Design Goal Our design goal is to develop a data-sharing framework for smart grid. It has the following desirable properties. 1) First, we propose a novel data-sharing framework for smart grid, where we combine the two popular infrastructures: a) the smart grid and b) cloud computing. In particular, we allow the electricity consumption reports generated in smart grid to be stored in the cloud, and the distributed ERs can obtain the statistics and analysis results from the cloud computing. Hence, our proposed framework can take advantages of cloud computing for smart grid. 2) Second, our proposed framework makes use of the homomorphic encryption technique to facilitate the statistics and analysis on the encrypted electricity consumption reports, and the proxy re-encryption technique to keep the statistics and analysis results secret from the cloud. 3) DaaS Model: The proposed framework takes advantages of the DaaS model in cloud computing, which can save a good amount of hardware and software maintenance cost for smart grid. Furthermore, the electricity consumption reports do not need to transmit over the network after the analysis request from the ER, which saves the communication cost in smart grid. 4) Privacy Preservation: The proposed framework should achieve privacy requirements of ECs. In particular: a) the electricity consumption reports stored in the cloud server cannot be revealed to anyone except the corresponding ER and b) the analysis results cannot be revealed to the one who has no corresponding analysis rights.

## III. PROPOSED DATA-SHARING FRAMEWORK

In this section, we present our data-sharing framework, which consists of four parts: 1) system initialization; 2) reports creation; 3) analysis grant; and 4) reports analysis. Before plugging into the framework detail, we first need to review the preliminaries, including bilinear groups, homomorphic encryption, and proxy re-encryption, which will serve as the basis of our proposed framework. *A. Preliminaries 1) Bilinear Groups:* Let G and G*T* be two multiplicative cyclic groups of prime order *q*. They are equipped with an admissible bilinear map $e : G \times G \rightarrow GT$, such that $e(ga1 , gb2 ) = e(g1, g2)ab$ for all $a, b \in Zq$ and any $g1, g2 \in G$. We denote BSetup as an algorithm that, on the input ofsecurity parameter *λ*, outputs the parameters *(*G,G*T , q, g, e)*, where $q \in \_(2\lambda)$. *2) Homomorphic Encryption:* Homomorphic encryption [15] is a special form of encryption that allows anyone with ciphertexts of messages *(m*1*, . . . ,mt)* to output a ciphertext of message *f (m*1*, . . . ,mt)* for some desired function *f* without knowing the decryption key. If the function *f* could be any function, then the homomorphic encryption is a fully homomorphic encryption. A concrete homomorphic encryption scheme is composed of the following four algorithms. 1) HE.KeyGen: The key generation algorithm is a randomized algorithm that takes a security parameter *λ* as input, and outputs the public/private key pair *(pk, sk)*. 2) HE.Enc: The encryption algorithm is a randomized algorithm that takes public key *pk* and a message *m* from the message space *M* as input, and outputs the ciphertext *c*. 3) HE.Dec: The decryption algorithm is a deterministic algorithm that takes the private key *sk* and a ciphertext *c* as input, and outputs the corresponding message *m*. 4) HE.Eva: The evaluation algorithm is a (possibly randomized) algorithm that takes the public key *pk*, a set of ciphertexts on messages *(m*1*, . . . ,mt)*, and a evaluation function *f* as input, and outputs the ciphertext *c* on *f (m*1*, . . . ,mt)*. The correctness of a homomorphic encryption scheme should satisfy the following two requirements for HE.KeyGen*(λ)* → *(pk, sk)*: HE.Dec*(sk,* HE.Enc*(pk,m))* = *m* and HE.Dec_*sk*, HE.Eva_*pk*, {HE.Enc*(pk,mi)*}*ti* =1, *f* __ = *f (m*1*, . . . ,mt)*. *3) Proxy Re-Encryption:* Proxy re-encryption [16] is a special kind of public key encryption, which allows a semitrusted proxy with some information to transform a ciphertext under one public key into another ciphertext under another public key. However, the corresponding message cannot be revealed during the transformation process. A concrete proxy re-encryption scheme is composed of the following five algorithms.

1) PRE.KeyGen: The key generation algorithm is a randomized algorithm that takes a security parameter λ as input, and outputs the public/private key pair (pk, sk). 2) PRE.Enc: The encryption algorithm is a randomized algorithm that takes public key pk and a message m from the message space M as input, and outputs the ciphertext c. 3) PRE.Dec: The decryption algorithm is a deterministic algorithm that takes the private key sk and a ciphertext c as input, and outputs the corresponding message m. 4) PRE.ReKey: The re-encryption key algorithm is a (possibly randomized) algorithm that takes one public/private key pair (pk1, sk1) and another public key pk2 as input, and outputs the corresponding re-encryption key rk. 5) PRE.ReEnc: The re-encryption algorithm is a (possibly randomized) algorithm that takes a ciphertext c1 under public key pk1, and a re-encryption key rk corresponding to the delegation from pk1 to pk2 as input, and outputs the ciphertext c2 under public key pk2.

## REFERENCES

[1] M. Jacobs, “13 of the largest power outages in history and what they tell us about the 2003 northeast blackout,” 2013. [Online]. Available: http://blog.ucsusa.org/mike jacobs/2003-northeastblackout- and-13-of-the-largest-power-outages-in-history-199

[2] D. Bobkoff, “10 years after the blackout, how has the power grid changed?” 2013. [Online]. Available: http://www.npr.org/2013/08/14/210620446/10-years-after-the-blackout-how-has-the-power-grid-changed

[3] R. Deng et al., “Sensing-performance tradeoff in cognitive radio enabled smart grid,” IEEE Trans. Smart Grid, vol. 4, no. 1, pp. 302–310, Mar. 2013.

[4] H. Liang, B. J. Choi, W. Zhuang, and X. Shen, “Towards optimal energy store-carry-and-deliver for PHEVS via V2G system,” in Proc. INFOCOM, Orlando, FL, USA, 2012, pp. 1674–1682.