Secure and privacy-preserving information aggregation for smart grids

Fengjun Li*

College of IST,
The Pennsylvania State University,
University Park, PA 16802, USA
E-mail: fli@ist.psu.edu
*Corresponding author

Bo Luo

Department of EECS,
The University of Kansas,
Lawrence, KS 66045, USA
E-mail: bluo@ku.edu

Peng Liu

College of IST,
The Pennsylvania State University,
University Park, PA 16802, USA
E-mail: pliu@ist.psu.edu

Abstract: In this paper, we present a distributed incremental data aggregation approach, in which data aggregation is performed at *all* smart metres involved in routing the data from the source metre to the collector unit. With a carefully constructed aggregation tree, the aggregation route covers the entire local neighbourhood or any arbitrary set of designated nodes with minimum overhead. To protect user privacy, homomorphic encryption is used to secure the data enroute. Therefore, all the metres participate in the aggregation, without seeing any intermediate or final result. In this way, our approach supports efficient data aggregation in smart grids, while fully protecting user privacy. This approach is especially suitable for smart grids with repetitive routine data aggregation tasks.

Keywords: smart grids; information aggregation; security; privacy-preserving.

Reference to this paper should be made as follows: Li, F., Luo, B. and Liu, P. (2011) 'Secure and privacy-preserving information aggregation for smart grids', *Int. J. Security and Networks*, Vol. 6, No. 1, pp.28–39.

Biographical notes: Fengjun Li did her BE in Automation from University of Sciences and Technology of China, and MPhil in Information Engineering from the Chinese University of Hong Kong. Currently she doing her PhD in Information Sciences and Technology at the Pennsylvania State University. Her areas of interest are data and application security, network security, privacy and healthcare informatics.

Bo Luo received his BE Degree from University of Science and Technology of China, MPhil Degree from the Chinese University of Hong Kong, and PhD Degree from Pennsylvania State University in 2008. He is presently an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Kansas. His research interest includes information and database security, information retrieval, Web-based multimedia technologies, and online social networks.

Peng Liu received his BS and MS Degrees from the University of Science and Technology of China, and his PhD Degree from George Mason University in 1999. He is a Professor of Information Sciences and Technology, Director of the Center for Cyber-Security, Information Privacy, and Trust, and Director of the Cyber Security Lab at Penn State. His research interests are in all areas of computer and network security. He has published a book and over 160 refereed technical papers. His research has been sponsored by DARPA, NSF, AFOSR, ARO, DHS, DOE, AFRL, NSA, TTC, CISCO, and HP. He is a recipient of the DOE Early Career Principle Investigator Award.

This paper is extended from a conference paper presented at the first IEEE Conference on Smart Grid Communications in 2010.

1 Introduction

Smart grids are envisioned by numerous and diverse stakeholders as the next-generation approach of delivering electricity to millions of households worldwide (Massoud Amin and Wollenberg, 2005; Lu et al., 2009; Johnson, 2010; Wolfs and Isalm, 2009; Son and Chung, 2009; Serizawa et al., 2010). The smart grids have introduced computation and communication capabilities into traditional power grids to make them 'smart' and 'connected'. Processing chips and storage units have been embedded into traditional electricity metres, so that they are capable of performing 'smart' functions. Then, smart metres communicate with electrical appliances at home as well as the generation and management facilities at the power companies, providing smart grids with great connectivity. Research and implementation on smart grids could be categorised at three layers (Sensing et al., 2008): "*smart generation, smart grid, and smart customer*". With the intelligent and networked metres, the smart grids enable instant monitoring of power delivery and consumption information, subscription of power usage and controlling from remote, advanced demand and outage management, usage management especially with respect to pricing (e.g., charging electrical cars at none-peak hours), etc. Therefore, it benefits end-users as well as power generation and distribution. Moreover, smart electricity metres could be further linked with smart water and gas metres to better coordinate and manage energy usage for smarter/greener homes (van Bruchem, 2006).

However, with all the advantages introduced by smart grids, security and privacy concerns start to arise (McDaniel and McLaughlin, 2009; Khurana et al., 2010):

1 hackers could compromise smart metres to manipulate power usage and energy costs

2 cyber-terrorists might fake power consumption data at a large scale to attack the power system, e.g., overloading nuclear power plants

3 attackers hacking into others' smart metres may control and tease with their electrical devices

4 adversaries might compromise smart metres, eavesdrop the communication, or hack into power company's database, to access power consumption data of the victim, from which they learn about the victim's daily activities, habits, and other privacy with reasonable inferences.

Many security and privacy vulnerabilities and threats have been studied in the research literature, however, most of the problems remain unanswered.

Instant aggregation of data and resources is an important function in smart grids (Sanders, 2010). For instance, aggregations of power usage data at multiple levels (neighbourhood, subdivision, district, city etc.) are conducted at different frequencies. Such information is essential for monitoring and predicting power consumption, allocating and balancing loads and resources, and administering power generation, etc. Therefore, it is of great importance to provide *efficient* and *secure* data aggregation in smart grids. To tackle the challenge, we present in-network aggregation for smart grids, in which aggregation is performed in a distributed manner, instead of centralising at the collector devices. To protect user and neighbourhood privacy in the aggregation, we employ homomorphic encryption to ensure that intermediate results are not revealed to any device enroute, while still maintaining an efficient and effective aggregation process.

The rest of the paper is organised as follows: in Sections 2 and 3, we introduce related works and background, especially homomorphic encryption. In Section 4, we present our problem and explore possible solutions. We conclude the paper in Section 5.

2 Related works

Research on smart grid spans a wide spectrum: from technology (Moslehi and Kumar, 2010; van Engelen and Stephanie Collins, 2010; Bose, 2010) to economy, marketing, policy and legal issues (Tabors et al., 2010; Schuler, 2010); from power generation, transmission, distribution (Wei et al., 2009; Saint, 2009), to load management, failure diagnosis and recovery (Masoum et al., 2010; Pasdar and Mirzakuchaki, 2009; Russell and Benner, 2010), to smart metre implementation and communications (Luan et al., 2009; Sood et al., 2009; Srinivasa Prasanna et al., 2009; Aggarwal et al., 2010; van Engelen and Stephanie Collins, 2010). Among these topics, we are particularly interested in security and privacy of smart grids. McDaniel and McLaughlin (2009) identifies several security and privacy vulnerabilities/threats in smart grids, and calls for attention and efforts from government, academia and industry. Khurana et al. (2010) reviews the security challenges in smart grids, with a special focus on trust, authentication and encryption. Metke and Ekl (2010a, 2010b) have articulated the security requirements for smart grid networks, and pointed out different security technologies to fulfill such requirements. Particularly, they have elaborated Public Key Infrastructure (PKI) and trustworthy computing, and the potential adoption in smart grid networks. As a comparison, we focus on a

particular problem – secure information aggregation in smart grids, instead of covering the broad area.

Various in-network data aggregation approaches have been proposed for sensor networks (e.g., Krishnamachari et al., 2002; Madden et al., 2002a), in which sensors are extremely restricted in computation and communication power due to their limited battery. In smart grid systems, although power of the smart metres is usually not a concern, communication bandwidth may still be insufficient, especially when frequent aggregation is desired. Meanwhile, sensors in the network are usually trusted to see the data from other sensors and the intermediate aggregation results, and most secure aggregation researches focus on defending against passive attacks (e.g., eavesdropper) (Castelluccia, 2005; Girao and Westhoff, 2005) or the attacks tampering with the aggregation mechanism using fake inputs (Chan et al., 2006; Frikken and Dougherty, 2008). However, in smart grids, power usage is considered as privacy of the owner, and should not be revealed to other metres. Therefore, traditional tree-based aggregation on plaintext does not apply.

In this paper, we employ homomorphic encryption to perform in-network aggregation but still keep the outputs and intermediate results secure. Homomorphic encryption is usually used for privacy-preserving operations (e.g., voting), in which operations are performed but operands (inputs) are not disclosed. Well-known homomorphic encryption schemes include: RSA, El Gamal (El Gamal, 1985), Paillier (Paillier, 1999), Naccache-Stern (Naccache and Stern, 1998) and Boneh-Goh-Nissim (Boneh et al., 2005), etc. In this work, *additive homomorphic* property is desirable for in-network data aggregation, therefore, we adopt Paillier cryptosystem (Paillier, 1999; Paillier and Pointcheval, 1999). To our best knowledge, there's no known homomorphic encryption scheme that provides full support of both addition and multiplication. In this sense, the best approach so far is the BGN cryptosystem (Boneh et al., 2005), which supports one multiplication between unlimited number of additions. Basically, it extends Paillier with bilinear groups. With one multiplication, the cipher text fall from cyclic group $\mathbb{G}$ to $\mathbb{G}_1$, and is still additively homomorphic on $\mathbb{G}_1$.

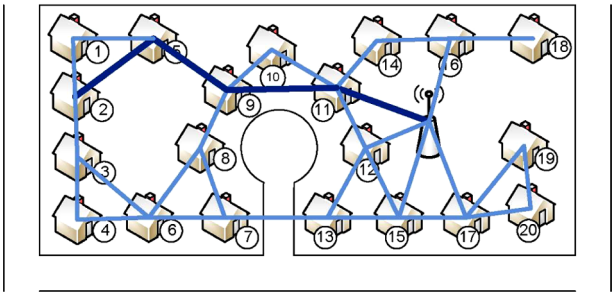
3 System model and design goals

In this section, we describe system models, followed by the design goals of the proposed aggregation protocol.

3.1 Network model

Although there have been different proposals on the smart grid communication infrastructure, the wireless-wired multi-layer architecture is the most popular approach, and has been adopted in some pilot projects. In this architecture, smart metres in a neighbourhood communicate with a *collector device* through a wireless mesh network. The collector device further communicates with central management unit of the grid operator through wired communication channels (e.g., LAN or dial-up) (McDaniel and McLaughlin, 2009; van Engelen and Stephanie Collins, 2010; Bose, 2010). With limited scope of coverage, the collector device may not be able to establish a direct connection with every smart metre in the neighbourhood, it is assumed every smart metre should have at least one communication path through a set of other smart metres in range to the collector device. Figure 1 shows an example of the communication infrastructure for a neighbourhood with 20 homes. In Figure 1, connection between metre 2 and the collector is routed through metres 5, 9, and 11.

Figure 1 An example of smart grid communication in a neighbourhood (see online version for colours)



3.2 In-network aggregation

A critical issue in current smart grids infrastructure is to accurately monitor usage data towards a better match between electricity production and consumption. Therefore, data aggregation (e.g., average/total power usage of a neighbourhood) becomes a very important type of query in smart grids (Sanders, 2010). In above network model, collector devices are expected to collect accurate and cumulative usage data from smart metres in their neighbourhood, and then calculate the aggregation results to report to the central management unit.

One natural approach is for each smart metre to establish a connection with the collector device, and use this channel exclusively to report its data to the collector. The collector then handles all the connections simultaneously, and condenses the data prior to sending it to the central management unit. Although simple and easy-adoptable, this approach introduces excessive network traffic as well as overwhelming demands at the collectors.

Example 1: Let us look at the example shown in Figure 1. For an aggregation query that covers the entire neighbourhood, it simultaneously establishes 20 connections to the collector. However, most of the connections are redundant. For example, connection between metre 11 and the collector can be reused by metre 2, 5 and 9.

Therefore, we propose an *in-network incremental aggregation approach*: when the collector is only interested in statistical measurements but not the individual readings, instead of requiring each metre to create an independent connection with the collector, we ask enroute metres to share the communication channel. Such type of aggregation is first proposed in sensor networks to reduce the amount of data transmitted within a wireless sensor network, where both the sensor nodes and aggregation nodes have restricted capacity and computation power (Girao and Westhoff, 2005; Castelluccia, 2005). Differently, we assume smart metres are not only measuring tools but also potential aggregators, which are more powerful than sensor nodes in both storage and computation capability.

In general, the in-network aggregation approach requires first to construct a virtual aggregation tree based on the network topology of the wireless mesh. In a bottom-up aggregation, each node in the tree collects data from its children, computes an aggregation over all the collected data and its own, and sends the aggregated result to its parent. The collector, as the root node, ultimately gets the aggregation result over the entire tree. Common aggregation operations include calculations of the mean or variance of the measured data, but sometimes applications of smart grids may require more complicate operations to facilitate smart pricing. At a broader scope, we define *in-network smart grids operations* as follows.

In-network smart grids operations

An operation at the collector device takes input from all (or a subset of) N smart metres: $f(I_1) \star f(I_2) \star \cdots \star f(I_i) \star f(I_N)$, where I_i is the input from smart metre i, $f()$ is a measuring function, and $\star$ is the operation. If the operation is associative and commutative, we are able to perform the operation of $\star$ in any arbitrary order, and deploy some of the computation to the smart metres in the network.

For example, the $\star$ operation in data aggregation could be *sum* (e.g., total utility usage in a neighbourhood) or *count* (e.g., the number of homes in the neighbourhood that are using a certain device). In *count* operation, the measuring function $f()$ is:

$$f(I_i) = \begin{cases} 1, & \text{the device at home } i \text{ is in use;} \\ 0, & \text{otherwise.} \end{cases}$$

3.3 Attack Model

In this paper, we assume all participants follow the honest-but-curious adversary model, a.k.a. semi-honest model (Goldreich, 2004). In this model, all parties are assumed to follow the protocol properly ('honest'); meanwhile, they keep all inputs from other parties and all intermediate computation results, from which they actively seek or infer knowledge about others ('curious'). Therefore, honest-but-curious adversaries keep the system functioning properly to avoid being identified by intrusion/abnormal detection mechanisms, while maximising the chance of obtaining others' privacy.

In our scenario, honest-but-curious smart metres do not tamper with the aggregation protocols: they do not spitefully drop or distort any source value or intermediate result; and they keep the system running smoothly. However, they will try to infer others' electricity usage by analysing messages and values that have been routed through them.

3.4 The problem

The aggregation results, even the partial results at intermediate smart metres, contain a big portion of user consumption information that are of interest of attackers. The aggregated data becomes targets of various attacks. Without protection, it is easy for a malicious attacker to passively eavesdrop the aggregation result at a particular smart metre or actively forge the aggregation value via compromised smart metres. Hence, it is critical to protect the *confidentiality* and *authenticity* of the aggregation results in smart grids.

Cryptographical schemes have been developed to secure communications among smart metres. In general, these security schemes can be classified into two categories: hop-by-hop and end-to-end secure aggregation schemes (Girao and Westhoff, 2005).

In hop-by-hop approaches, while data is encrypted from being recovered by eavesdroppers, intermediate smart metres are allowed to decrypt the collected data, apply aggregation functions, and then re-encrypt the aggregation results prior to sending them to the next-hop smart metres. Hence, hop-by-hop aggregation requires extra decryption and encryption on each smart metre, and thus introduces excessive computation effort. Moreover, hop-by-hop aggregation introduces a severe privacy problem. The utility usage information carries a significant amount of user privacy, and thus needs to be protected from attackers, within (e.g., malicious/curious smart metres) and outside (e.g., passive eavesdroppers) the smart grid systems. However, in secure hop-by-hop aggregation, the mathematical operations for aggregation tasks cannot be performed over commonly encrypted ciphertext, each participating smart metre still needs to see intermediate aggregation results routed through itself in plaintext.

Therefore, we turn to secure end-to-end aggregation approaches, which are considered more efficient in computation and providing better confidentiality than hop-by-hop aggregation approaches. Secure end-to-end aggregation was first proposed by Girao et al. for concealed sensor data aggregation in Girao and Westhoff (2005). Security infrastructures based on privacy homomorphism (PH) are adopted to prevent even the aggregating intermediate nodes from reading the measured plaintext data. However, most of these approaches are based on symmetric PH protocols, and thus cannot be directly applied in securing smart grids

aggregations, in which smart metres play the roles as both measuring tools and aggregation nodes.

3.5 Design goal

Our design goal is to design a secure and privacy-preserving data aggregation protocol to tackle the privacy issue in secure end-to-end data aggregation and defend against honest-but-curious smart metres from reading the intermediate aggregation results. Specifically, our design goal includes:

- *Security*: The aggregation results should be protected against passive eavesdroppers at any position of the smart grids.
- *Privacy*: The aggregation results should be protected against honest-but-curious smart metres that participate in the aggregation.
- *Efficient*: The algorithm should be efficient in terms of communication overhead. The computation overhead should also be reasonable due to the restricted computation power of smart metres.
- *Generality*: It is unlikeable to design a separate scheme for each aggregation purpose, the aggregation scheme should apply to various aggregation functions to meet the requirements of smart pricing.

4 Secure and privacy-preserving data aggregation protocol

In this section, we present our secure and privacy-preserving data aggregation protocol. We first give an overview of the protocol and then present its details.

4.1 Protocol overview

To design the secure and privacy-preserving aggregation protocol, we first construct an aggregation tree that efficiently connect all the smart metres in the neighbourhood to the collector device. To be efficient, the aggregation tree should be shallow (i.e., the height of the tree is reasonably small) and less-bushy (i.e., the number of children of each node is also reasonably small). Then, we employs a particular semantically secure homomorphic encryption algorithm – Paillier cryptosystem, and construct operation plans for each smart metres in the aggregation protocol. Measured electricity usage data on each participating smart metres is encrypted in a way that algebraic operations of the plaintext are allowed to be performed on the cipher domain to enable aggregation functions.

Next, we present the protocol details, which includes three phases: *aggregation tree construction*, *operation plan construction*, and *aggregation operations*.

4.2 The aggregation tree

The goal of in-network aggregation is to perform aggregation function in a hierarchical manner: at each intermediate node the aggregation is performed over the readings of all the nodes downstream from it. Therefore, it is important to construct the hierarchy – the aggregation path that covers all the smart metres in the neighbourhood. For each aggregation task, all or a subset of the nodes on the aggregation path are selected to participate in the task.

In the network model described in Section 3.1, the aggregation path is constructed within a wireless mesh network of one collector device and N wireless smart metres. Therefore, we can consider the smart metre network as a graph $G(V, E)$, where V is the set of smart metres (as vertices) and E is the set of available wireless links (as edges) between any two smart metres. Intuitively, the aggregation tree is a *spanning tree* of the graph, which consists of a (minimal) subset of E that connects all vertices in a hierarchical structure. Moreover, in order to include all the vertices in the tree, the graph should be connected – every smart metre should have at least one communication path to the collector device.

For a given graph, there may exist multiple spanning trees of different structures. It is critical to identify a spanning tree that best fits our expectations, while relatively easy to be constructed. In our scenario, the aggregation tree should always root at the collector node, which initialises all the aggregation tasks and collects the final results. Meanwhile, for a network with N smart metres (excluding the collector device), every aggregation tree will consist of $N + 1$ nodes and N edges. The aggregation is recursively calculated in a bottom-up manner: every node in the aggregation tree takes inputs from itself and its children nodes; it then aggregates the data and sends the result to its parent node.

There are two major concerns on constructing the aggregation tree:

1. the height of the tree should be small in order to reduce the maximum hops in the longest aggregation path, thereby reduce the end-to-end aggregation time
2. for an interior node of the tree that performs intermediate aggregation, it should not have too many children in order to avoid excessive computation and communication load.

To achieve the first goal, the spanning tree is constructed by a breadth-first traversal of the graph. The tree is constructed in a way that the collector node is selected as the root, and then all other nodes are connected to root via the shortest path route. The implementation of such breadth-first traversal tree construction algorithm is simple. Similar as proposed in Madden (2002b), the root initiates a tree construction message with its own identifier, and the distance (hops) to root set to 0.

When receiving such construction message, a node updates its construction message with the shortest-path neighbour by setting this neighbour as its parent and assigning its distance to root as the one of its parent's distance plus 1. In this way, the height of the tree is the same as the shortest distance from the furthest node to the collector.

However, the constructed aggregation tree via breadth-first traversal is shallow and bushy. Nodes at higher levels always have larger number of children. In the case that a node has too many children, it is more likely to become a bottleneck in the aggregation. Therefore, the aggregation tree should be balanced to avoid potential bottlenecks. It is ideal to balance the tree in a way that a node cannot have more than k children. It becomes the degree-constrained spanning tree problem in graph theory that the maximum vertex degree is limited to a preset constant k. It is a NP-complete problem, so we propose a heuristic algorithm to efficiently *re-balance* the tree:

- Starting from the root, we check the vertex degree of every node to see if it is larger than the preset threshold k;
- For a node T with too many children, we first check if any child of T is also connected to a less-populated neighbours. If Yes, we move this child (with the subtree) to the less-populated neighbour with shortest path to root. Continue
- If T still has too many children, continue moving until the vertex degree of T decreases below k.

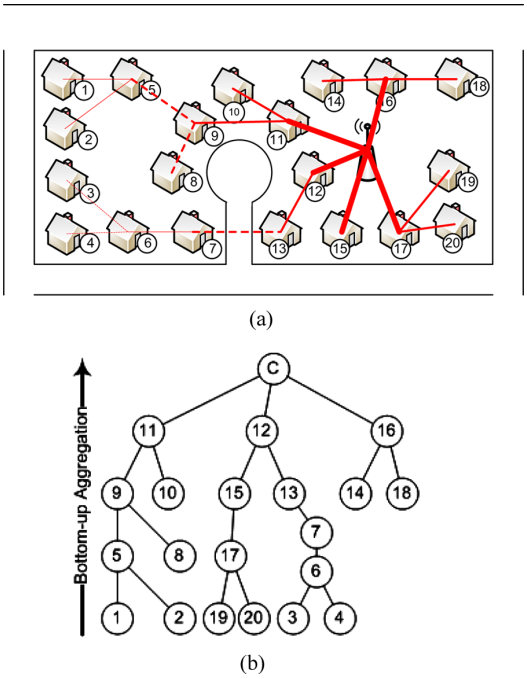
The re-balancing may cause the height of the tree to increase. Although the construction process may take a while, it is still efficient since in most cases, the the collector device has the network graph of the entire neighbourhood and thus it constructs the aggregation tree without probing all the smart metres. In addition, since the network routing in the smart metre network is relatively static, once constructed, an aggregation graph will remain stable for an extended period of time.

Example 2: Figure 2(a) shows an example of the breadth-first traversal of the graph shown in Figure 1, while Figure 2(b) shows the resulting aggregation tree. Assume the maximum vertex degree is set to 3, obviously, the collector should not be allowed to handle five children simultaneously. We re-balance the tree by first moving node 15 to node 12, and then moving subtree at node 17 to node 15.

4.3 Homomorphic encryption

To provide security, the aggregated results should be encrypted from being seen by other parties expect the collector. Meanwhile, to protect privacy, not only the data involved in the aggregation process should be encrypted, but also the aggregation operations should be performed over such encrypted data. The fundamental basis for such secure and privacy-preserving aggregation protocols are cryptographic algorithms with privacy homomorphism (PH) property, called homomorphic encryption algorithms. It represents a group of semantically secure encryption functions that allow certain algebraic operations on the plaintext to be performed directly on the ciphertext. Mathematically, given a homomorphic encryption function $\mathsf{E}()$, and two messages $x, y \in \mathbb{Z}_N$, we are able to compute $\mathsf{E}_k(x \star y) = \mathsf{E}_{k_1}(x) \circ \mathsf{E}_{k_2}(y)$, without knowing the plaintext x, y or the private key. In practice, $\star$ can represent either addition or multiplication operations.

Figure 2 An example of aggregation tree construction: (a) breadth-first traversal of the network graph and (b) the aggregation tree constructed from the traversal



Generally speaking, *additive homomorphic* property is more desirable for in-network data aggregation in smart grids, considering the typical aggregation tasks mainly involve combinations of utility readings. Therefore, we adopt Paillier cryptosystem (Paillier, 1999; Paillier and Pointcheval, 1999) in this work, which is one of the two commonly used additive homomorphic encryption functions (the other one – the BGN cryptosystem (Boneh et al., 2005) – is an extension of Paillier with bilinear groups).

We briefly explain the Paillier cryptosystem as follows:

Key generation

- Pick two large prime numbers p and q.
- $N = p \cdot q$ and $\lambda = l\ cm(p-1, q-1)$, where l cm represents least common multiple.
- Select a random number g where $g \in \mathbb{Z}^*_{n^2}$.

- Set function $L(u)$ as: $L(u) = (u-1)/N$.
- Ensure that N divides the order of g: check if $L(g^{\lambda} \mod N^2)$ and n are co-prime, i.e. $\gcd(L(g^{\lambda} \mod N^2), N) = 1$.
- (N, g) is the public key.
- (p, q) is the private key.

Encryption

- We want to encrypt the message: $m \in \mathbb{Z}_N$.
- Select a random number: $r \in \mathbb{Z}_N^*$.
- Encrypt m using: $c = \mathsf{E}(m) = g^m \cdot r^N \mod N^2$.

Decryption

- We want to decrypt ciphertext: $c \in \mathbb{Z}_{N^2}^*$
- Decrypt with: $m = \mathsf{D}(c) = (\frac{L(c^{\lambda} \mod N^2)}{L(g^{\lambda} \mod N^2)}) \mod N$.

As we can see, Paillier cryptosystem is additively homomorphic. Given $c_1 = \mathsf{E}(m_1)$ and $c_2 = \mathsf{E}(m_2)$, $\forall m_1, m_2 \in \mathbb{Z}_N$, we have the following homomorphic properties on $\mathbb{Z}_N$:

- *Addition*: Sum of plaintext can be calculated by multiplication of ciphertext: $\mathsf{D}(c_1 \cdot c_2 \mod N^2) = m_1 + m_2 \mod N$. Similarly: $\mathsf{D}(c_1 \cdot g^{m_2} \mod N^2) = m_1 + m_2 \mod N$
- *Multiplication*: Product of plaintext and a constant k can be calculated over the cipher text: $\mathsf{D}(c_1^k \mod N^2) = k \cdot m_1 \mod N$.
- However, given two ciphertext, there is no known way to compute the product of corresponding plaintext without decryption.

Besides additive homomorphism, Paillier cryptosystem has a nice property due to the existence of the random blinding factor r: the encryption is indeterministic, i.e., using different r, the same message will be encrypted to difference cipher. Hence, Paillier cryptosystem is resistant to dictionary attacks.

4.4 Aggregation operation plan

After constructing the aggregation tree, the collector node can initiate aggregation tasks by disseminating *operation plans* through the tree. The operation plans are constructed for each participating node, and deployed to each smart metre in a top-down manner.

Particularly, an operation plan is a 7-tuple: $\{T_{ID}, Trigger, Data, Collect, Operation, Destination, Key\}$, where

- T_{ID} is an arbitrary but unique identifier used to identify the message.
- $Trigger$ defines at what time the aggregation starts. It can be periodically at a certain frequency, or upon the connector's request, or at a particular time.
- $Data$ defines what information will be collected from the local smart grid in the aggregation. For example, the current power usage reading, or the electricity usage for charging electric vehicles in a 24-hour period.
- $Collect$ tells a smart metre a specific set of nodes whose input is waited to collect, e.g., its children in the aggregation tree.
- $Operation$ tells a smart metre what operation to be performed, including pre-processing, encryption and operations for aggregation. Since homomorphic encryption will be used at each participating smart metre, the collector also needs to translate operations on the plaintext to operations on the ciphertext to define $Operation$. For instance, additions on the plaintext will be converted to multiplications on the ciphertext in the Paillier cryptosystem.
- $Destination$ denotes the parent node, i.e., to whom the output from $Operation$ will be sent.
- And finally, Key is a set of keys to be used for encryption. In Paillier cryptosystem, Key is a public key of the collector that is used to encrypt the local data at each smart metre.

Please note that not all the fields in the operation plan are mandatory. For instance, when the aggregation only covers a subset of the entire graph, the $Data$ field is blank for the metres that do not participate in the aggregation. For one-time aggregation that is conducted instantly, $Trigger$ field could be omitted and the aggregation starts right upon receiving the operation plan. Also, when $Trigger$ specifies a particular time, we define it as denoting the time of local data reading instead of the time of aggregation. This guarantees, in time-sensitive tasks, no mis-synchronisation will be caused due to the latency in computing or network communication.

Since we adopt Paillier cryptosystem, Key represents the public key (N, g) of the collector device that is publicly known by all the smart metres, and $E_K()$ denotes the ciphertext encrypted with public key Key.

Example 3: With the aggregation tree in Example 2, to calculate the total output power (in KW) at time t_0 in the entire neighbourhood, the aggregation plan at node 9 is constructed as: $\{tid, t_0, \texttt{power}, \{N_5, N_8\}, E_K(power) \times f(I_5) \times f(I_8), N_{11}, K\}$, where $f(I_5)$ and $f(I_8)$ are the encrypted inputs of node 5 and 8, respectively.

When a smart metre in the network receives the operation plan, it follows the following protocol:

1 The smart metre determines if it should start the aggregation immediately, or wait for the trigger.

2 When an aggregation is to be performed, the smart metre retrieves local data as requested in $Data$ field of the plan, and encrypts it with Key as local input.

3 Then the smart metre waits for the inputs from the child nodes, as defined in $Collect$. Once receiving all the inputs, it follows $Operation$ to perform aggregation over all the input ciphertext.

4 Finally, the smart metre sends the output from Step 3 to the $Destination$ node, i.e., its parent node in the aggregation tree. The output message is constructed as $\{T_{ID}, TS, Data\}$, where TS is the timestamp of local data retrieval in Step 2. This timestamp is used for synchronising different occurrences of repeating tasks.

To explain the execution of operation plan, we continue with Example 3 and demonstrate the operations at node 9 and the collector device in the following example.

In the following examples, C_{p_i} denotes the ciphertext of the local reading of node i, and C_{o_i} denotes the final output from node i.

Example 4: When node 9 receives the aggregation plan $\{tid, t_0, \texttt{power}, \{N_5, N_8\}, E_K(power) \times f(I_5) \times f(I_8), N_{11}, K\}$, it first retrieves its own `power` reading at time t_0 and encrypts the reading with K to get the local input $E_K(P_9)$. From the $Collect$ field, node 9 knows it should wait for the input from node 5 and 8, $f(I_5)$ and $f(I_8)$. Once receiving $f(I_5)$ and $f(I_8)$, node 9 calculates its aggregation result $f(I_9) = E_K P_9 \times f(I_5) \times f(I_8)$, as described in $Operation$, and submits $f(I_9)$ to node 11.

In the aggregation, the collector node ultimately receives inputs from nodes 11, 12 and 16, and computes the aggregation result as $f(I_{11}) \times f(I_{12}) \times f(I_{16})$. Let us denote the ciphertext of the local reading of node i as $C_{p_i} = E_K(P_i)$, and the final output from node i as $C_{o_i} = f(I_i)$. After further decomposing, we have the aggregation result at the collector as:

$$\begin{aligned} C_{\text{col}} &= C_{o_{11}} \times C_{o_{12}} \times C_{o_{16}} \\ &= (C_{p_{11}} \times C_{o_9} \times C_{o_{10}}) \times (C_{p_{12}} \times C_{o_{13}} \times C_{p_{15}}) \\ &\quad \times (C_{o_{14}} \times C_{o_{18}}) \\ &= C_{p_{11}} \times C_{p_9} \times C_{p_5} \times C_{p_1} \times C_{p_2} \times \cdots \times C_{p_{28}}. \end{aligned}$$

Finally, the collector decrypts C_{col} to obtain the aggregation result $D(C_{\text{col}}) = P_{11} + P_9 + P_5 + P_1 + P_2 + \cdots + P_{20} = \sum_{i=1}^{20} P_i$.

4.5 Aggregation operations

One of our design goal is to provide generality to incorporate as many aggregation functions as possible into a same representation space. Therefore, we extend the definition in Section 3.2 to define *in-network weighted-aggregation* operation.

In-network weighted-aggregation

An aggregation at the collector device takes input from all N smart metres: $f(I_1) \star f(I_2) \star \cdots \star f(I_i) \star f(I_N)$, where I_i is a multi-valued input from smart metre i, $f()$ is a weighted measuring function $f(I_i) = c_i \times I_i$, and $\star$ is the operation.

Let $c_i = 1$ for all is, and I_i choose a single-valued partial input, e.g. power usage at smart metre i, the weighted-aggregation becomes the simple power usage aggregation operation in Example 4. However, with flexibly designed c_i and I_i, weighted aggregation can be used to perform various typical aggregation tasks.

Weighted Pricing: In smart grids, the grid operator applies weighted pricing on categorised power usage. One typical data aggregation is to measure the aggregated cost under current pricing strategy, e.g., electricity consumed during busy hour is more expensive in unit price than the one in off-busy hours. Each household has readings for both categories, therefore, the input from each smart metre is a tuple $< I_{i1}, I_{i2} >$. The measuring function f becomes $f(I_i) = c_{i1} \times I_{i1} + c_{i2} \times I_{i2}$. With homomorphic properties for addition and constant multiplication, the operation $\star$ is interpreted as ${C_{p_{i1}}}^{c_{i1}} \times {C_{p_{i2}}}^{c_{i2}}$.

5 Security analysis

5.1 Passive attacks

The Paillier cryptosystem adopted in in-network aggregation approach is semantically secure (IND-CPA): polynomial time adversary who intercepts the communication cannot derive significant information about the plaintext from the ciphertext and the public key. Meanwhile, with the existence of the random blinding factor r, the same data will be encrypted to different cipher with different r, which makes the approach resilient to the dictionary attack.

5.2 Malleability

All homomorphic encryption systems are malleable – given the cipher and public key, an adversary could generate another cipher which decrypts to another meaningful plaintext in the same domain as the original plaintext. As a result, a dishonest or fake smart metre could falsify the data, which causes inaccurate aggregation result. However, this problem is not introduced by in-network aggregation, considering a dishonest smart metre could falsify its data in any aggregation approach. The problem could be solved by increasing the physical and software security of smart metres and improving authentication. Although we do not consider false data injection attack in this paper, detecting manipulation of the aggregate by the adversary is part of our future work.

5.3 Discussion: abnormal data detection

Abnormal event detection is a very important task in smart grid operations. Abnormal detection includes abnormal data subscription/publication, data aggregation, abnormal operations, intrusion detection, etc. Generally speaking, in data-related operations, there are two types of abnormal data:

1 fake data injected by intruders, which is usually spiteful, may not follow any pattern, and is comparably harder to detect

2 abnormal data generated by mal-functioning devices, which is usually repeating or follows a pattern, and is relatively easier to detect.

In this paper, we make the assumption of honest-but-curious smart metres, hence, detecting the first type of abnormal data is outside of our scope. Users interested in this subject may refer to Liu et al. (2009); Kosut et al. (2010) for further information. Meanwhile, we briefly discuss the detection of the second type of abnormal data in secure information aggregation approach.

Mal-functioning devices such as smart metres and smart appliances introduce abnormal data into the smart grid communication system. It is also called 'pollution'. In our application, mal-functioning devices reports inaccurate data, which affects the aggregation results, and may further influence the central control facility on energy production and/or pricing. Fortunately, such mal-functioning devices are comparably easier to detect, since they repeatedly report wrong data, and the reported data is often altered in orders of magnitudes compared with normal data. We introduce an approach that is somewhat similar to binary search for abnormal node detection. In a typical setting, the outlier could be detected with the following protocol:

1 The collector device examines the aggregated data, and determines if it is malicious. This could be achieved through rule-based reasoning or statistical analysis.

2 If the collector device determines that the aggregation result has taken inaccurate inputs, it will repeat the aggregation task, but include only $1/2$ of the nodes of the previous aggregation. This will be achieved by enforcing a constant c_i in the *Operation* field of the aggregation request. Hence, the input from smart metre i becomes $c_i \times f(I_i)$. Note that c_i is sent to node i in encrypted form, and the '$\times$' operation is conducted on cipher texts (employing homomorphic properties). By manipulating c_i, the collector device is able to select a subset of the nodes and the nodes are kept 'blind'.

3 The collector device repeats step (1) to determine if the faulty node has been included in this aggregation, and repeat step (2) with the subset of nodes that contains the mal-functioning node (either the aggregated half or the other half of the nodes). Repeat until the outlier is detected. In this way, when there is one mal-functioning node in the system of n nodes, we will be able to detect it in $\log(n)$ iterations. Note that, when there exist multiple mal-functioning nodes, we need to do two aggregations in step (2), since both subsets may contain mal-functioning nodes. However, the worst case computation is still bounded at $O(k \times \log(n))$, where k denotes the number of mal-functioning nodes.

This detection method is simple and relatively efficient. Meanwhile, as an advantage of the homomorphic encryption based aggregation, the detection is non-disclosed to the involved smart metres – they do not know that they have been suspected and checked. In this way, if an adversary repeatedly forges its data, it will be detected before it notices that it is targeted. On the other hand, it relies on the collector device to recognise polluted aggregation results, which can be difficult if the pollution is minor (fortunately, minor pollution in aggregation data is usually less harmful to the power system).

The general abnormal detection problem in smart grid systems is difficult. It involves two key steps:

1 determine if the data is polluted (or poisoned)

2 identifying the sources of the pollution.

Both steps require intensive research efforts, and the problem becomes even more difficult in the presence of deliberate attacks. We plan to further analyse this issue and tackle the problem in our future works.

6 Evaluation

Now we compare the complexity of the in-network aggregation approach (with homomorphic encryption) presented in this paper to the traditional aggregation approach, which collects the input from every smart metre and performs the aggregation at the collector device. We will compare from several dimensions including network traffic, system scalability, system robustness, security and privacy, and the overall computation.

Network: In the traditional approach, messages from all smart metres are routed to the collector device simultaneously. The average number of hops for each message to be transmitted to the collector device, $\overline{h}$, is determined by the size of the neighbourhood (the residential area covered by a particular collector), the wireless communication range of each smart metre, and the routing scheme. Assume the number of nodes in the graph is N. To transmit data from all the smart metres participating in the aggregation, the total load on the network will be $\overline{h} * N$ hops. However, in the proposed

in-network aggregation approach, the total load will be N hops.

Example 5: In the example neighbourhood, if we use the traditional aggregation approach to collect data from all the smart metres, the total load on the network will be 50 hops. On the other hand, the in-network aggregation approach only needs 20 hops in total. By choosing in-network aggregation, we saved 60% of network load. In real world cases, a collector device often covers a large neighbourhood, which indicates a large $\overline{h}$, and therefore more savings on network load.

Scalability, bottleneck and robustness: As we have shown, the overall system scalability highly depends on the smart metre network topology. In a well-designed network, the aggregation tree is usually wide and shallow, which makes our approach very scalable. The longest path in an aggregation process is the graph diametre, which usually grows at a speed of $\sqrt{N}$. Also, since most of computation are distributed to smart metres, with re-balance scheme, there is almost no unavoidable bottleneck in the in-network aggregation approach. On the contrary, most of the computation in the traditional approach are centralised at the collector device. Considering decrypting messages from all the smart metres is highly computation-intensive, the collector becomes the major bottleneck, especially when the number of smart metres in the neighbourhood (N) gets large.

In in-network aggregation, when one smart metre fails, the failure will be detected immediately by its parent in the aggregation tree and reported to the collector. Then, the collector will update the aggregation tree and re-issue the query. The recover process will fail only when the aggregation graph becomes unconnected (e.g., split into two isolated subgraphs), which is often caused by the failure of a large number of nodes. Obviously, in such cases, most approach will also fail.

Computation: The choice of encryption/decryption scheme has a strong impact on the computation at both smart metres and the collector. In particular, asymmetric encryption is more computationally expensive than symmetric encryption (e.g., AES and triple-DES). Here, we compare the computation load at both smart metres and the collect in two approaches. In the traditional approach, with no in-network aggregation, each smart metre will encrypt its message once with the public key, while the collector needs to decrypt N messages. In the proposed approach, every smart metre needs to encrypt the message once with homomorphic encryption (still asymmetric encryption), but the collector device only needs to apply one asymmetric decryption (to the final aggregation result). Moreover, the in-network aggregation approach distribute the computation of aggregation at the collector (e.g., the addition on the plaintext) to intermediate smart metres, and introduces extra overhead (e.g., the multiplication on the ciphertext). However, such overhead is small and acceptable per smart metre. For instance, a smart metre with k children in the aggregation tree only needs to perform $k+1$ multiplications for each aggregation, where k is usually small (controlled by the re-balance process).

7 Conclusion and future works

In this paper, we present in-network data aggregation for smart grids. In this approach, a spanning tree rooting at the collector device is constructed to cover all of the smart metres. Aggregation is performed in a distributed manner in accordance to the aggregation tree – each node collects data from its children, aggregates them with its own data, and sends the intermediate result to the parent node. Homomorphic encryption is employed to protect the privacy of the electricity usage information, so that inputs and intermediate results are not revealed to smart metres on the aggregation path, while the aggregation is still correctly performed.

In this paper, we have assumed honest but curious model for the smart metres. However, there could be adversaries that maliciously forge their own data to manipulate the aggregation results. Such adversaries and false data reports need to be detected through advanced auditing approaches, which is one of our ongoing research. Meanwhile, we also plan to further refine the algorithm and deploy it in a real world smart grid system.

Acknowledgements

This work was supported by AFOSR FA9550-07-1-0527 (MURI), ARO W911NF-09-1-0525 (MURI), NSF CNS-0905131, NSF CNS-0916469, and a University of Kansas General Research Fund (GRF: 2301420).

References

Aggarwal, A., Kunta, S. and Verma, P.K. (2010) 'A proposed communications infrastructure for the smart grid', *Innovative Smart Grid Technologies*, January, Gaithersburg, MD, pp.1–5.

Boneh, D., Goh, E. and Nissim, K. (2005) 'Evaluating 2-dnf formulas on ciphertexts', *Proceedings of Theory of Cryptography (TCC)*, Interlaken, Switzerland, pp.325–341.

Bose, A. (2010) 'Smart transmission grid applications and their supporting infrastructure', *IEEE Transactions on Smart Grid*, Vol. 1, No. 1, June, pp.11–19.

Brown, R.E. (2008) 'Impact of smart grid on distribution system design', *IEEE Power and Energy Society General Meeting – Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, pp.1–4.

Castelluccia, C. (2005) 'Efficient aggregation of encrypted data in wireless sensor networks', *MobiQuitous*, *IEEE Computer Society*, Washington DC, USA, pp.109–117.

Chan, H., Perrig, A. and Song, D. (2006) 'Secure hierarchical in-network aggregation in sensor networks', *CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, ACM, Alexandria, Virginia, USA, pp.278–287.

Frikken, K.B. and Dougherty IV, J.A. (2008) 'An efficient integrity-preserving scheme for hierarchical sensor aggregation', *Proceedings of the First ACM Conference on Wireless Network Security*, New York, NY, USA, pp.68–76.

El Gamal, T. (1985) 'A public key cryptosystem and a signature scheme based on discrete logarithms', *Proceedings of CRYPTO 84 on Advances in Cryptology*, Springer-Verlag, Inc., New York, pp.10–18.

Girao, J. and Westhoff, D. (2005) 'Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks', *IEEE International Conference on Communications (ICC '05)*, Seoul, Korea.

Goldreich, O. (2004) *Foundations of Cryptography: Volume II (Basic Applications)*, Cambridge University Press, New York, NY.

Johnson, A.P. (2010) 'The history of the smart grid evolution at southern california edison', *Innovative Smart Grid Technologies*, January, Gaithersburg, MD.

Khurana, H., Hadley, M., Lu, N. and Frincke, D.A. (2010) 'Smart-grid security issues', *Security Privacy, IEEE*, Vol. 8, No. 1, pp.81 –85.

Kosut, O., Jia, L., Thomas, R.J. and Tong, L. (2010) 'Malicious data attacks on smart grid state estimation: attack strategies and countermeasures', *Proc of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, pp.220–225.

Krishnamachari, B., Estrin, D. and Wicker, S.B. (2002) 'The impact of data aggregation in wireless sensor networks', *Proceedings of the 22nd International Conference on Distributed Computing Systems*, Washington DC, USA, pp.575–578.

Liu, Y., Reiter, M.K. and Ning, P. (2009) 'False data injection attacks against state estimation in electric power grids', *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS '09*, ACM, New York, NY, USA, pp.21–32.

Lu, J., Xie, D. and Ai, Q. (2009) 'Research on smart grid in China', *Transmission Distribution Conference Exposition: Asia and Pacific*, October, Seoul, Korea, pp.1–4.

Luan, S-W., Teng, J-H., Chan, S-Y. and Hwang, L-C. (2009) 'Development of a smart power meter for AMI based on ZigBee communication', *PEDS*, Taipei, Taiwan, pp.661–665.

Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W. (2002a) 'Tag: a tiny aggregation service for ad-hoc sensor networks', *SIGOPS Oper. Syst. Rev.*, Vol. 36, Special Issue, pp.131–146.

Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W. (2002b) 'Tag: a tiny aggregation service for ad-hoc sensor networks', *OSDI*, Boston, MA.

Masoum, M.A.S., Moses, P.S. and Deilami, S. (2010) 'Load management in smart grids considering harmonic distortion and transformer derating', *Innovative Smart Grid Technologies*, Gaithersburg, MD, pp.19–21.

Massoud Amin, S. and Wollenberg, B.F. (2005) 'Toward a smart grid: power delivery for the 21st century', *Power and Energy Magazine, IEEE*, Vol. 3, No. 5, pp.34–41.

McDaniel, P. and McLaughlin, S. (2009) 'Security and privacy challenges in the smart grid', *Security Privacy, IEEE*, Vol. 7, No. 3, pp.75 –77.

Metke, A.R. and Ekl, R.L. (2010a) 'Security technology for smart grid networks', *IEEE Transactions on Smart Grid,*, Vol. 1, No. 1, June, pp.99–107.

Metke, A.R. and Ekl, R.L. (2010b) 'Smart grid security technology', *Innovative Smart Grid Technologies*, January, Gaithersburg, MD.

Moslehi, K. and Kumar, R. (2010) 'Smart grid – a reliability perspective', *Innovative Smart Grid Technologies*, January, Gaithersburg, MD.

Naccache, D. and Stern, J. (1998) 'A new public key cryptosystem based on higher residues', *CCS '98: Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, CA, pp.59–66.

Paillier, P. (1999) 'Public-key cryptosystem based on composite degree residuosity classes', *Proceedings of Eurocrypt '99*, Berlin, Heidelberg, pp.223–238.

Paillier, P. and Pointcheval, D. (1999) 'Efficient public-key cryptosystems provably secure against active adversaries', *Advances in Cryptology – Proceedings of Asiacrypt '99*, Springer-Verlag, London, UK, pp.165–179.

Pasdar, A. and Mirzakuchaki, S. (2009) 'Three phase power line balancing based on smart energy meters', *EUROCON*, Saint-Petersburg, Russia, pp.1876–1878.

Russell, B.D. and Benner, C.L. (2010) 'Intelligent systems for improved reliability and failure diagnosis in distribution systems', *IEEE Transactions on Smart Grid*, Vol. 1, No. 1, June, pp.48–56.

Saint, B. (2009) 'Rural distribution system planning using smart grid technologies', *IEEE Rural Electric Power Conference*, Fort Collins, Colorado, pp.B3–8.

Sanders, W.H. (2010) 'Progress towards a resilient power grid infrastructure', *Proceedings of the IEEE Power & Energy Society General Meeting*, July, Minneapolis, Minnesota, pp.1–3.

Schuler, R.E. (2010) 'Electricity markets, reliability and the environment: smartening-up the grid', *43rd Hawaii International Conference on System Sciences*, Honolulu, HI, pp.1–7.

Sensing, A., Bose, A. and Wittig, W. (2008) 'Power system design: basis for efficient smart grid initiatives', *IET Seminar Digests*, Vol. 2008, No. 12380, pp.58–58.

Serizawa, Y., Ohba, Y. and Kurono, M. (2010) 'Present and future ICT infrastructures for a smarter grid in Japan', *Innovative Smart Grid Technologies*, January, Gaithersburg, MD.

Son, S-Y. and Chung, B-J. (2009) 'A Korean smart grid architecture design for a field test based on power IT', *Transmission Distribution Conference Exposition: Asia and Pacific*, October, Seoul, Korea, pp.1–4.

Sood, V.K., Fischer, D., Eklund, J.M. and Brown, T. (2009) 'Developing a communication infrastructure for the smart grid', *Electrical Power Energy Conference (EPEC), 2009 IEEE*, October, Montreal, QC, pp.1–7.

Srinivasa Prasanna, G.N., Lakshmi, A., Sumanth, S., Simha, V., Bapat, J. and Koomullil, G. (2009) 'Data communication over the smart grid', *IEEE International Symposium on Power Line Communications and Its Applications*, Dresden, Germany, pp.273–279.

Tabors, R.D., Parker, G. and Caramanis, M.C. (2010) 'Development of the smart grid: Missing elements in the policy process', *43rd Hawaii International Conference on System Sciences*, Honolulu, HI, pp.1–7.

van Bruchem, H. (2006) 'Think smart! The introduction of smart gas meters', *23rd World Gas Conference*, Amsterdam, pp.1–8.

van Engelen, A.G. and Stephanie Collins, J. (2010) 'Choices for smart grid implementation', *HICSS'10*, Honolulu, HI, pp.1–8.

Wei, X., Yu-hui, Z. and Jie-lin, Z. (2009) 'Energy-efficient distribution in smart grid', *Sustainable Power Generation and Supply, 2009. SUPERGEN '09. International Conference on*, Nanjing, China, pp.1–6.

Wolfs, P. and Isalm, S. (2009) 'Potential barriers to smart grid technology in Australia', *Australasian Universities Power Engineering Conference*, September, Adelaide, SA, pp.1–6.