Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops

Secure and Efficient Capability-based Power Management in the Smart Grid

Dongwon Seo and Heejo Lee
Div. of Computer and Communication Engineering
Korea University
Seoul, Korea
{*aerosmiz, heejo*}*@korea.ac.kr*

Adrian Perrig
CyLab
Carnegie Mellon University
Pittsburgh, PA, USA
adrian@ece.cmu.edu

***Abstract*—As a smart grid is becoming a promising technology to control and save power generation and consumption, smart grid security should be a preliminary consideration to prevent from catastrophic failures. Especially, excessive power consumption can be a significant issue, because power provider cannot react quickly to such massive demand that can cause blackouts through wide regions. Many studies, such as DDoS prevention schemes, have been done to solve excessive resource consumption for the legacy networks (e.g., the Internet). However, power management in the smart grid needs its own requirements: reliable power supply, privacy preservation, efficient data communication and malicious behavior detection. Existing smart grid schemes consider some of the requirements, but do not address all the requirements. In order to satisfy the four requirements, we propose a secure and efficient power management mechanism leveraging a homomorphic data aggregation and capability-based power distribution. The proposed mechanism enables to gather the power demands of customers securely and efficiently, and to distribute power to customers who have the capability. Furthermore, each customer can verify whether one's request is correctly delivered to the utility, and each distributor can detect misbehaving customers exceeding their capabilities. From our evaluation, we show that a power provider consumes 11.12 seconds until power distribution. It is a tolerably short time for a power provider to endure against excessive power consumption. Through this paper, we proposes the first concept of secure and efficient power management in the smart grid.**

***Keywords*-Smart grid security; capability-based power management; homomorphic aggregation**

I. Introduction

As electricity demands are increasing, smart grids have been developed to save energy, reduce cost and increase reliability and transparency. Moreover, many countries are trying to demonstrate the feasibility of a smart grid system [1]–[3]. As the smart grid is in charge of providing and controlling power, it is no doubt that the failure of the smart grid can cause a catastrophe that will take more damage than current grid system failures, such as the New York city blackout [4]. As the result, many organizations have been striving for developing secure smart grid systems [5]–[7].

Since controlling power generation and consumption is one of the major objectives for smart grids, smart grids have to provide reliable power in any situation. Especially, excessive power consumption can cause serious damages to power generation and distribution, since a power generation cannot support quickly for sudden massive demands. The current Internet is already experiencing such issues like DDoS attacks and flash crowds. Both situations cause excessive resource consumption (e.g., network bandwidth), so that users cannot access specific services. Similarly, excessive power consumption can happen in smart grids. That is, an adversary can compromise and control smart appliances to cause excessive power consumption. Note that there are surprisingly many malicious codes for smart phones despite their short histories, and compromised phones are capable of DDoS attacks via 3G and WiFi networks [8]. Smart grids are possibly exposed to such attacks causing catastrophic failures [9].

Thus, smart grids need to manage power consumption, and several power management schemes exist: load shedding by demand response (DR) [10], rolling blackout [11] and distributed generation [12]. Load shedding is only for legitimate users who obey power control policies (e.g., real-time pricing). Malicious users have no reason to obey the policies. Rolling blackout is to block power to where the demand for electricity exceeds the power supply capability; therefore, it causes collateral damages to legitimate users nearby malicious ones. Distributed generation can be a temporal solution against such malicious power shortages, because it provides insufficient power to cover wide areas. Existing schemes cannot solve the situation that attackers maliciously cause excessive power consumption.

As compared to the existing power managements for smart grids, a capability-based approach for the Internet can complement the drawbacks, because its major goal is to reserve resources for legitimate users [13]–[15]. However, simply adopting a capability-based approach is not a solution due to unique requirements for smart grids. Since individual power demands are very sensitive information [5], [16], [17], power utilities (power providers) should preserve individual privacy. Furthermore, utilities need to handle simultaneous power demands from numerous customers, and, most of all, malicious behaviors should be detectable in order to prevent from system failures.

Now four main requirements for smart grid power management can be defined as follows:

- R1 (Reliability): To provide reliable power to legitimate users under excessive power demands
- R2 (Efficiency): To communicate efficiently (less overhead) for large networks
- R3 (Privacy): To preserve individual privacy
- R4 (Security): To detect malicious behaviors

To best our knowledge, no existing power management scheme for smart grids addresses the requirements. In this paper, we propose a secure and efficient power management mechanism leveraging a capability-based approach and secure data aggregation. The capability-based approach guarantees resources (power in this paper) for authorized users. Secure data aggregation using an asymmetric homomorphic encryption satisfies privacy-preserving and efficient

978-0-7695-4429-8/11 $26.00 © 2011 IEEE
DOI 10.1109/ISPAW.2011.36

data forwarding. In our proposed mechanism, a child (e.g., a customer) sends his demand encrypted by a root's homomorphic public key to his parent (e.g., an aggregator or a distributor). Then, a parent aggregates children's demands using homomorphic aggregation and forwards the aggregated demands to his parent. At the end, a root (e.g., a power provider) receives the total power demands. Since the demands are encrypted by the root's homomorphic public key, only the root can decrypt and obtain the total demands. After that, each node verifies the aggregation result using the concept of a hash tree [18] to determine whether or not the aggregation result is correct.

If the aggregated demand exceeds power supply, the root needs to scale down the amount of power supply. The root decides the scale of power supply depending on the demands; for example, if the root only supplies the half of the demand, the scale becomes 0.5. Then the root sends information, including the scale and the homomorphic private key to its children. Each child decrypts his aggregated demand using the private key and decides the amount of power supply for his children using the scale. Each child performs these operations (decrypts and decides the power amount) until reaching to a grandparent of a customer. A grandparent does not send the private key to his children (parents of customers) in order not to reveal customer's privacy (power demand). Instead, the grandparent only sends the scale to his children, so that the parents of customers can decide the total power amount (the total capabilities) for his customers. A customer's demand constrains the amount of the customer's capability, so that his parent can detect if he misbehaves such as consuming more than his demand. Section IV explains the details of each phase: bootstrapping, aggregation, verification and distribution.

The contribution of our work is twofold:

1) We propose the first concept of capability-based power management for smart grids. It enables to provide reliable power to legitimate users based on capabilities under excessive power consumption; moreover, the demand aggregation enables efficient data forwarding to the power provider.
2) We design a mechanism to satisfy not only power distribution, but also privacy preservation and misbehaving detection. It enables to verify the aggregation and to detect misbehaving customers who demands more than their capabilities.

The remainder of this paper is organized as follows. Section II, we briefly explain the concepts of smart grids and data aggregation, which form the basis for our approach. Section III presents our problem definition and assumptions. Section IV explains how our proposed mechanism works with a sample scenario. In section V, we analyze and discuss potential threats and countermeasures. Section VI introduces several problems that the proposed mechanism does not address. Numerical measurements for the proposed mechanism's time overhead is presented in Section VII. Section VIII discusses existing approaches. Finally, Section IX presents our conclusion.

II. Background

In this section, we briefly describe the background of capability-based DDoS defense and homomorphic encryption, which are the key mechanisms of our proposed mechanism.

A. Capability-based DDoS Defense

The purpose of capability-based DDoS defense is to preserve bandwidth for legitimate hosts so that legitimate hosts are guaranteed to communicate under DDoS attacks. Capability-based DDoS defenses use cryptographic techniques to verify the capabilities at network routers. Each capability includes expire information so that the routers can decide whether or not to forward the packets related to the capability. There are three types of packets: priority packets, request packets and best-effort packets. A sender has to transmit the request packet to obtain the capability from the receiver. If the receiver grants the capability, then the sender is allowed to send priority packets. When there is link congestion, capability-based routers forward the priority packets first. Best-effort packets are sent by legacy hosts.

B. Homomorphic Encryption

A homomorphic encryption scheme allows arithmetic operations to be performed on ciphertexts, and there are two types of homomorphic encryptions: additive and multiplicative homomorphic encryptions. Since we compute the aggregation result by adding the customer's demand, we adopt the Paillier cryptosystem [19], which is commonly used for additive homomorphic encryption.

Here, we describe the concept of the Paillier cryptosystem. The details for key generation, encryption and decryption are shown in [19] (refer to [20] for simpler explanation). Let $E(\cdot)$, m and r be the encryption function, a message and a random number, respectively ($m \in \mathbb{Z}_N$ and $r \in \mathbb{Z}_N^*$). Given m, the cipertext, c, is $c = E(m) = g^m \cdot r^N \ mod\ N^2$, where g is a random number for the base ($g \in \mathbb{Z}_{N^2}^*$), and N is the multiplication of two large prime numbers . Then, the additive homomorphic property is as follows:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= (g^{m1} \cdot r_1^N)(g^{m2} \cdot r_2^N)\ mod\ N^2 \\ &= g^{m1+m2}(r_1 r_2)^N\ mod\ N^2 \\ &= E(m_1 + m_2) \end{aligned}$$

Moreover, the Pailler cryptosystem satisfies *semantic security* using r, which means that we cannot distinguish $E(m_1)$ from $E(m_2)$ as $N \to \infty$. Therefore, the encrypted message, m, is indeterministic, and attackers cannot guess m from $E(m)$.

III. Problem Definition

In this section, we define the problems when existing capability-based approaches are adopted to smart grids. Also, we explain the assumptions and attacker models in smart grids.

A. Power Management Problem in Smart Grids

Smart grids handle with the very sensitive resource, electricity. Of the conventional resource management schemes against excessive consumption (e.g., DDoS) in the Internet [21], [22], the capability-based approach shows the remarkable performance for guaranteeing resources for legitimate users, and it is the reason that we decide to adopt the capability-based approach to smart grids [13], [14]. Since all of them were designed to operate for the Internet, it is

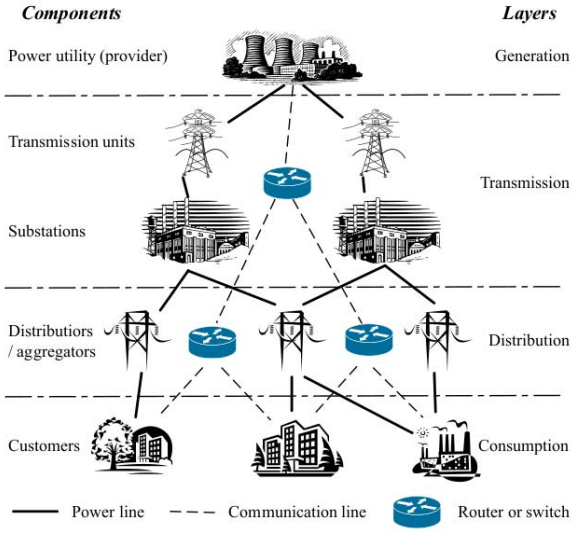

Figure 1. An example of the smart grid topology: It forms a tree-like topology as the parent nodes are usually connected to multiple children.

infeasible to naively adopt the existing approaches regardless of the smart grid environment. The following describes the issues inherited from smart grid features, and each issue is associated to each requirement that we mentioned in Section I.

Reliability (R1). The smart grid network has to provide reliable power supply to legitimate customers, since electricity is the most important and sensitive resource in our lives [5], [23].

Efficiency (R2). The smart grid network forms a tree-like topology as shown in Fig. 1. A node in a higher layer, termed a parent (e.g., a power utility), generally supports multiple nodes in a lower layers, termed a child (e.g., substations). Thus, a parent needs to handle many requests from his children, and an efficient communication method is required [16].

Privacy (R3). Power usage of a customer can inform many facts; how many people are living in, what kinds of appliances they are using, etc. Therefore, in the smart grid, a customer does not want to reveal his usage [5], [16], [17].

Security (R4). Smart grids are very attractive attack targets, because attackers can achieve economic gains and cause the catastrophic damages that are incomparable to the Internet damages. Thus, a defense mechanism should be able to detect and prevent potential attacks [5], [24].

The power management problem in smart grids is to devise a scheme that satisfies the four requirements.

B. Assumptions

Throughout this paper, we set several assumptions from the perspective of topology, communication channels, key managements and reliable nodes. These assumptions are technical bases in order that the proposed mechanism works.

Logical topology. In practice, multiple power and communication lines can exist among the nodes, so that the physical topology can form a mesh network. However, we assume that a parent is generally connected to multiple children while a child is connected to few parents (a tree-like topology). For example, a power utility is linked to multiple substations, while the substations belong to a single power utility like shown in Fig. 1.

Smart grids can use two separate lines: the power line and the communication line. In this paper, we assume that the two lines logically form the same topology, which is the tree-like topology.

No congestion in communication channels. There is no congestion in communication channels (e.g., DDoS attacks in the Internet), so that all data via communication lines can reach to the destination.

Secure communication channels. Communication channels are protected by a pairwise symmetric key between parents and children ($K_{parent,child}$).

Key managements. A power utiltity deploys and manages two asymmetric key pairs (a public key : a private key): for homomorphic aggregation ($PK_a : SK_a$) and for singing to the capability ($PK_s : SK_s$).

Trusted power utility. Any nodes (e.g., aggregators or customers) can conduct malicious behaviors except power utilities.

C. Attack Models

Considering the vulnerabilities classified by NIST [5], we define several attack models in terms of passive and active attacks.

- Passive attack: Attackers do not forge any data but listen and gather the data silently. It includes eavesdropping for collecting private information.
- Active attack: Attackers generate and/or forge data locally and remotely, so that they attempt to find some security flaws and to take benefits. It includes result-tampering attacks, excessive power consumption and false data injection.

The brief explanation of each attack is as follows.

A1: Eavesdropping. Attackers monitor and steal private information via wire and/or wireless communications. These kinds of attacks can be solved by using cryptographic protocols, and we also assume secure communication channels throughout this paper.

A2: Result-tampering attacks. Attackers send false aggregation results to their parents/children in order to cause their misbehaviors. Since we use a capability-based approach leveraging data aggregation, attackers can forward false aggregation results in order to cause power shortage or redundancy.

A3: Excessive power consumption. Attackers consume excessive power in order to cause overhead to the utility. Since we use a capability-based approach, attackers can launch this attack by consuming more power than their capabilities.

A4: False data injection. Attackers may inject malicious data into compromised smart meters to misestimate their information, such as power usage. This attack can be detected by state estimation schemes [25], [26].

A1 and A4 are out of our scope since they can be protected by existing schemes, such as cryptographic protocols

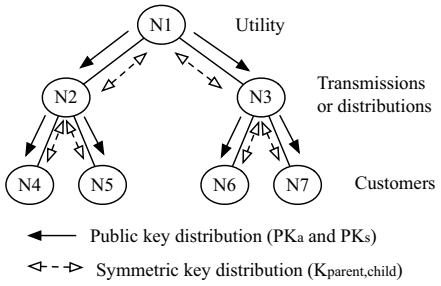

Figure 2. The bootstrapping phase

and state estimation schemes. However, A2 and A3 can cause significant failures in smart grids by misusing the proposed mechanism. Thus, we focus on A2 and A3 attacks. In Section V, we analyze the possibilities of the two attacks in detail.

IV. Proposed Mechanism Description

Our design principle is derived from the four issues in Section III-A: reliability, efficiency, privacy and security issues. For the reliability issues, we adopt a capability-based approach that customers use tickets (capabilities) to receive power supply. For the efficiency issues, a data aggregation is used. For example, distributors/aggregators in Fig. 1 receive and aggregate the power demands of their children, and then forward the aggregated data to their parents (e.g., substations). For the privacy issues, we utilize a homomorphic encryption, which is the Paillier cryptosystem [19]. It allows to blindly aggregate children's data without revealing their real values. For the security issues, we verify the aggregation result and the customer ticket (capability) using a hash tree [18] and a hash chain, respectively.

Our proposed mechanism consists of four phases: bootstrapping, aggregation, verification and distribution phases. The remainder of this section describes how the phases work based on a simple example.

A. Bootstrapping Phase

In the bootstrapping phase, a utility broadcasts two public keys, PK_a and PK_s. PK_a is used for homomorphic encryption and aggregation of the customer's demand, and PK_s is used for decrypting digitally signed information by a utiltiy. Also, each parent/child pair shares a symmetric key $K_{parent,child}$ in order to establish a secure channel.

The procedure of the bootstrapping phase is as follows (viz., Fig 2).

1) $N1$ broadcasts PK_a and PK_s to all nodes.
2) Each parent/child pair exchanges a symmetric key.
3) When $N1$ suffers from excessive power consumption, $N1$ sends messages to his children that informs the start of capability-based power management during t, and the messages eventually spread to all nodes, where t denotes the time period meaning how long the utility uses the capability-based approach (e.g., 1 hour).

B. Aggregation Phase

After a customer ($N4$) is informed the "capability start" message from his parent ($N2$), $N4$ sends $K_{N2,N4}\{PK_a\{V_{N4}\}\}$ to $N2$, where V_{N4} denotes the power demand of $N4$, PK_a denotes the utility's homomorphic public key, and $K_{N2,N4}$ denotes the symmetric key between $N2$ and $N4$. Since the customer encrypts using the utility's homomorphic public key, the parent cannot decrypt to reveal the customer's power demand, but the parent can aggregate $PK_a\{V_{N4}\}$ with the other customer's demand, say $PK_a\{V_{N5}\}$. Then, the parent obtains $PK_a\{V_{N2}\} = PK_a\{V_{N4}\} + PK_a\{V_{N5}\} = PK_a\{V_{N4} + V_{N5}\}$. Eventually, the utility ($N1$) can obtain the demands of whole customers, $PK_a\{V_{N1}\} = \sum_{n=1}^{\#customers} PK_a\{V_k\} = PK_a\{\sum_{n=1}^{\#customers} V_{Cn}\}$. Now, only the utility is capable of decrypting $PK_a\{V_{N1}\}$ using his homomorphic private key SK_a.

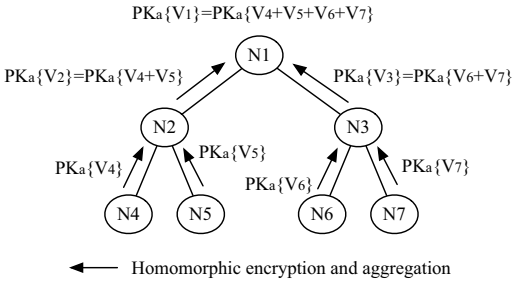

Figure 3. The aggregation phase

The procedure of the aggregation phase is as follows (viz., Fig 3). Note that we omit the symmetric key notations (e.g., $K_{N1,N2}\{...\}$) in the Fig 3, even though each child encrypts his demand using his symmetric key.

1) $N4$ sends his encrypted power demand $PK_a\{V_4\}$ to $N2$. The other leaves also send their encrypted power demands to their parents.
2) $N2$ aggregates children's power demands from $N4$ and $N5$; $PK_a\{V_2\} = PK_a\{V_4 + V_5\} = PK_a\{V_4\} + PK_a\{V_5\}$. Similarly, $N3$ also obtains the aggregation $PK_a\{V_3\}$ based on $PK_a\{V_6\}$ and $PK_a\{V_7\}$.
3) $N1$ receives $PK_a\{V_2\}$ and $PK_a\{V_3\}$, and finally obtains the demands of whole leaves, $PK_a\{V_1\} = PK_a\{\sum_{k=4}^{7} V_n\} = \sum_{k=4}^{7} PK_a\{V_k\}$.

C. Verification Phase

Through the aggregation phase, the utility achieves the demands of all customers. Now, each customer needs to verify whether his demand is successfully aggregated and delivered to the utility. We adopt the concept of the Merkle hash tree to verify the aggregation. To describe the concept easier, we borrow the term, "off-path nodes", from [18].

Off-path nodes. This is the set of nodes in a tree that is the set of all the siblings of each of the nodes on the path from a specific node to the root of the tree.

For example, in Fig. 4, $N4$'s off-path nodes are $N5$ and $N3$, because the path from $N4$ to $N1$ is $N4 \rightarrow N2 \rightarrow N1$, and $N5$ and $N3$ are the siblings of the path. Each node can verify the utility's aggregation result by comparing the hashed aggregation result with the hashed values of

the off-path nodes. Since we use the homomorphic aggregation, we can take the benefit that $H(PK_a\{V_{N1}\}) == H(\sum_{k=1}^{\#customers} PK_a\{V_k\})$.

We describe the procedure of the verification phase with the example of Fig 4.

1) $N1$ obtains the hashed aggregation result $H(PK_a\{V_1\})$ using a hash function, such as SHA-1. Then $N1$ sends $SK_s\{H(PK_a\{V_1\})\}$ and $PK_a\{V_3\}$ to $N2$. Note that $H(PK_a\{V_1\})$ is digitally singed by the $N1$'s private key SK_s so that no one except $N1$ creates $SK_s\{H(PK_a\{V_1\})\}$.
2) $N2$ decrypts $SK_s\{H(PK_a\{V_1\})\}$ using the $N1$'s public key PK_s. And, $N2$ obtains $H(PK_a\{V_1\})$.
3) $N2$ performs $H(PK_a\{V_2\} + PK_a\{V_3\})$ and compares it with $H(PK_a\{V_1\})$. If matched, $N2$ knows that his demand is successfully aggregated.
4) $N2$ sends $SK_s\{H(PK_a\{V_1\})\}$, $PK_a\{V_3\}$ and $PK_a\{V_5\}$ to $N4$.
5) $N4$ decrypts $SK_s\{H(PK_a\{V_1\})\}$ using the $N1$'s public key PK_s. And, $N4$ obtains $H(PK_a\{V_1\})$.
6) $N4$ performs $H(PK_a\{V_4\} + PK_a\{V_3\} + PK_a\{V_5\})$ and compares it with $H(PK_a\{V_1\})$. If matched, $N4$ knows that his demand is successfully aggregated.
7) $N4$ informs the verification result (correct or incorrect) to $N1$.

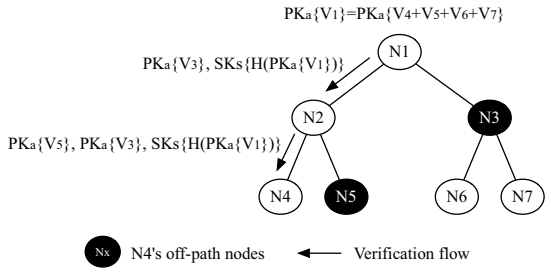

Figure 4. The verification phase

D. Distribution Phase

The distribution phase is divided to two subphases: power allocation and capability verification. Power allocation is that each parent forwards the information to decide the power amount for each child. Capability verification is to verify whether or not the customer consumes power using proper capabilities.

Power allocation. After the verification phase, the utility decides the scale of power generation depending on the aggregated demand. For instance, the aggregated demand is 1,000 kw/h; however, if the utility is capable of generating 500 kw/h, then the utility sets the scale as 0.5.

After deciding the scale (S), the utility informs the homomorphic private key SK_a and the scale S to his children. Note that SK_a and S are digitally signed by the utility, SK_s. Each child can decrypt his aggregation $PK_a\{V_k\}$ using SK_a, and decide the allocated power by $S \cdot V_k$. Then, each child forwards SK_a and S to his children. However, if a parent of a customer receives SK_a, the parent can decrypt the customer's demand. Therefore, SK_a should be forwarded until a grandparent of a customer. Distributing the homomorphic private key until grandparents of customer can cause a privacy exposure if a grandparent is a malicious node. However, in practice, the nodes nearby a utility are under direct control of the utility, so that the utility easily verifies whether or not grandparents are compromised.

Fig. 5 describes the procedure of the power allocation.

1) $N1$ decides S, and sends $V_2, SK_s\{SK_a\}$ and $SK_s\{S\}$ to $N2$.
2) $N2$ decrypts $PK_a\{V_2\}$ using SK_a and compares the decrypted result to V_2 from $N1$. $N2$ computes his allocated power P_2 by $S \cdot V_2$.
3) $N2$ sends V_4 and $SK_s\{S\}$. In this case, $N2$ does not send $SK_s\{SK_a\}$, because it can reveal the demands of customers ($N8$ and $N9$).
4) $N4$ decrypts $PK_a\{V_4\}$ and compares the decrypted result to V_4 from $N2$. $N4$ computes his allocated power P_4 by $S \cdot V_4$.

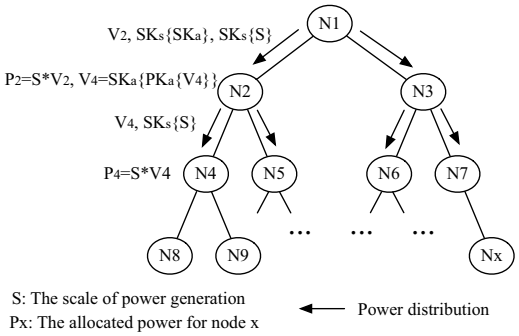

Figure 5. The power distribution phase

Capability verification. This subphase is only performed between parents (e.g., distributions) and leaves (e.g., customers). Since all parents of leaves determine their allocated power amount, a parent needs to decide how much amount of power he should provide to his children (customers). To keep customer privacy, a parent needs to provide power without knowing the demand of each customer.

Therefore, we use the concept of a ticket as the capability of a customer. A parent gives "scissors" to a child, and the scissors cut a child's demand into several pieces. The pieces become the tickets for the child. The parent takes the ticket and verifies the correctness of the ticket. In the proposed mechanism, the scissors become the "ticket generation rate", and the correctness of the ticket is verified by a "hash chain".

To help understand, Fig. 6 shows an example. Assume that $N8$ (the customer)'s demand is 100 kw/h, and the ticket generation cycle (the scissors) are 5 kw. It means that $N8$ has to send a ticket to $N4$ (the parent) at every 5 kw consumption. The following shows the procedure of the capability verification.

1) $N4$ decides the ticket generation rate, 5 kw, and notifies the rate to $N8$.
2) $N8$ obtains 20 tickets ($\frac{100}{5}$), and selects two nonces, i and j.
3) $N8$ notifies the expire condition of the ticket (Exp) and the verification code ($Veri$). Exp is made by the hashed value of each hashed value of the nonce,

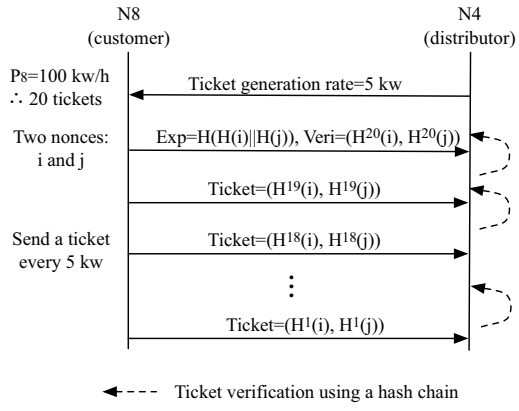

Figure 6. Capability verification

$H(H(i)||H(j))$. $Veri$ is made by the last hash chaining values of each hashed value of the nonce $[H^{20}(i),\ H^{20}(j)]$, where $H^n(i)$ denotes the hash chain of length n.

4) $N8$ sends a ticket $[H^{19}(i),\ H^{19}(j)]$ to $N4$ after consuming 5 kw.
5) $N4$ hashes each values, so that $N4$ can obtain $[H(H^{19}(i)),\ H(H^{19}(j))]$. $N4$ verifies the received ticket by comparing $[H(H^{19}(i)),\ H(H^{19}(j))]$ with $[H^{20}(i),\ H^{20}(j)]$.
6) $N4$ finally sends the last ticket, $[H^{20}(i),\ H^{20}(j)]$.
7) $N8$ stop providing power to $N4$.

The reason that the use of two nonces is to prevent from guessing the hash chain length by the parent. If a customer uses a single nonce, i, then a parent can guess the number of customer's tickets by chaining $Exp = H(i)$. However, as the customer uses two nonces to create Exp, the parent cannot guess the number of customer's tickets, because $H^n(H(i)||H(j)) \neq H(H^n(i)||H^n(j))$.

V. Security Analysis

As mentioned in Section III-C, we focus on the attacks targeting result-tampering attacks (A2) and excessive power consumption (A3).

A. Result-tampering Attacks

Attackers intentionally do not add the demands of specific children to cause lack of power supply for the children. However, through the verification phase, each node can verify the aggregation result.

In Fig. 3, for example, assume that $N2$ does not add the demand of $N4$, then $H(PK_a(V_1))$ becomes $H(PK_a(V_5) + PK_a(V_6) + PK_a(V_7))$. In order that $N2$ wants to deceive $N4$, $N2$ should find $PK_a(X)$, where X can result in $H(PK_a(V_1)) == H(PK_a(X) + PK_a(V_3) + PK_a(V_5))$. Note that $PK_a(V_3) = PK_a(V_6) + PK_a(V_7)$. Otherwise, $N4$ recognizes the false aggregation. Thus, $PK_a(X)$ should be $PK_a(V_3 + V_5 - V_4)$. To do that, $N2$ needs to send $PK_a(V_3+V_5-V_4)$ to $N4$; however, the Paillier's cryptosystem does not provide the subtractive homomophic property. Therefore, attackers cannot find $PK_a(X)$ that is equal to $PK_a(V_3 + V_5 - V_4)$.

B. Excessive power consumption

To consume excessive power, attackers can spoof the ticket to gain more power than their capabilities. For example, in Fig. 6, $N8$ can send $Veri = [H^{30}(i),\ H^{30}(j)]$ instead of $[H^{20}(i),\ H^{20}(j)]$. Then, he can obtain power 50 kw more, because he creates 10 tickets more. However, this attack can be easily detected since $N4$ knows the total amount of power for his children. If someone uses power more than his capability, $N4$ alerts the excessive consuming situation to the root. Then, the root decrypts the demand of the customer, and is able to detect the misbehaving child based on the demand. Since our scheme takes a time to detect excessive consumption, a parent needs to reserve some amount of power. The reserved power will be provided to legitimate customers until the root cuts out the attacker. By adjusting the scale S, a parent can decide the amount of reserved power.

Interestingly, if there are more number of attackers or greedier attackers, faster detection is possible. That is, faster consumption by attackers causes faster consumption of the total amount, and a parent identifies the abnormal situation faster.

VI. Open Problems

There are several problems that the proposed mechanism does not address. We would like to discuss them as open problems.

A. Instant Detection for Excessive Capabilities

Even though the proposed mechanism can detect the false capability, it takes some time until a parent recognizes the abnormal situation. However, it is required that we design to detect the abnormal situation instantly, because the electricity is a very sensitive resource. To do that, a ticket should have stronger relationship with the customer's demand $PK_a(V)$ so that an attacker cannot forge the ticket after the aggregation phase.

B. Multiple Power Providers

In practice, there are multiple power providers; thus, the provider of one's house can be different from the neighbors. In this case, we can logically divide the topology into several trees depending on the number of utilities, and customers use symmetric/asymmetric keys send messages corresponding to their utilities. Also, merging the capabilities between different utilities can be an issue in the case of multiple providers.

VII. Numerical Measurements for Time Overheads

In this section, we set several variables to measure the time taken for each phase, and approximate time overheads of the proposed mechanism.

A. Experimental assumptions

We assume several variables related to a topology, encryption/decryption/hash performance and network delay.

- Topology: It forms a tree that the depth is d.
- Encryption/decryption/hash performance: A symmetric encryption/decryption takes α millisecond (ms), and an asymmetric encryption/decryption takes β ms. A hash operation takes γ ms.

- Network delay: This represents physical link delay for a hop. We assume that network delay for each hop is k ms.

B. Time Overhead for Each Phase

Since we assume that key deployments for bootstrapping phase, we exclude the time measurement for the bootstrapping phase. We measure the time overhead for the aggregation, verification and distribution phases.

The time overhead for the aggregation phase. We measure the time that a provider (root) receives a customer's (leaf's) demand. The time overhead can be measured as follows.

1) A customer encrypts his demand using asymmetric homomorphic encryption: β ms.
2) An intermediate node performs symmetric encryption/decryption for secure communication: $2\alpha(d+1)$ ms (because there are $d+1$ nodes between a provider and a customer).
3) A provider decrypts the customer's demand using asymmetric homomorphic encryption: β ms.

Therefore, the time overhead for the aggregation phase is $2\alpha(d+1)+2\beta+d\cdot k$ ms, where $d\cdot k$ denotes the network delay for d hops.

The time overhead for the verification phase. We measure the time to complete the verification phase.

1) A customer receives and decrypts the values of off-path nodes $PK_a\{V_3\}$ and $PK_a\{V_5\}$ (viz., Fig. 4): k ms.
2) The customer verifies the signed value $PK_a\{V_1\}$: β ms.
3) The customer hashes the sum of off-nodes values and his own value: γ ms.
4) The customer informs the verification result to a provider using symmetric encryption: k ms.

Since the off-nodes values are delivered via d hops (from the provider to the customer), the time overhead becomes $d(k+2\alpha+\beta+\gamma)+d\cdot k$ ms, where 2α denotes the symmetric encryption/decryption overhead for each node.

The time overhead for the distribution phase. We measure the time that a parent of a customer determines his allocated power amount.

1) An intemediate node verifies the homomorphic private key $SK_s\{SK_a\}$ (viz., Fig 5): β ms.
2) The node verifies the scale $SK_s\{S\}$: β ms.
3) The node sends the signed homomorphic private key and the signed scale: k ms.
4) These operations are repeatedly performed until a grandparent of a customer $N2$ receives the private key and the scale. Therefore, it takes $(d-2)(k+2\beta)$ ms.
5) A grandparent only sends the scale to his child (a parent of a customer): k ms.
6) The child $N4$ verifies the scale: β ms.

Therefore, the time overhead for the distribution phase becomes $(d-2)(k+2\alpha+2\beta)+(k+\beta)$.

According to [27], [28], we can infer the time overhead for a crypto hash function (SHA-1), a symmetric encryption/decryption algorithm (AES-256) and an asymmetric encryption/decryption algorithm (Paillier encryption), which are 6.25 ms, 12 ms and 200 ms, respectively. These results are obtained by Intel 1.8∼2.4GHz CPU computers, which are appropriate machines for aggregators or distributors in smart grids. Also, we set $d = 15$ and $k = 20$ since they are average hop counts and network delay for the current Internet.

Table I
TIME OVERHEAD FOR EACH PHASE.

Phase	Aggregation	Verification	Distribution
Formula	$2\alpha(d+1)$ $+2\beta+d\cdot k$	$d(k+2\alpha+\beta+\gamma)$ $+d\cdot k$	$(d-2)(k+2\alpha+2\beta)$ $+(k+\beta)$
Time overhead	1,084 ms	4,053 ms	5,992 ms

Table I shows how much time each phase takes. The proposed mechanism totally takes 11.12 seconds from demand aggregation to power distribution. It is a tolerable time duration for the power provider in order to initialize a power management mechanism.

VIII. RELATED WORK

Our work is motivated from Li *et al.*'s research [16] that proposed a secure aggregation mechanism for smart grids. The authors showed how to aggregate customer's data with privacy preservation using a homomorphic encryption, and we extend the research in order to satisfy secure verification and distribution using the capability-based approach, even though attackers attempt to exploit the mechanism. Bartoli *et al.* also proposed another aggregation protocol for smart grids [29]. The secure aggregation protocol provides lossless and end-to-end security in Machine-to-Machine (M2M) wireless communication. It focuses on the communication among smart devices rather than the smart grid framework itself. Besides, homomorphic encryption methods are applied to secure e-voting mechanisms [30], [31] that requires to blindly collect ballots.

In sensor networks, many researches have presented secure aggregation mechanisms. Chan *et al.* [18] presented a secure hierarchical in-network aggregation scheme. The authors improved existing aggregation schemes in order to design robust aggregation that works against malicious result-tampering. It disseminates verification codes consisting of off-path values to each sensor nodes. Mlaih *et al.* [32] presented a secure aggregation scheme in wireless sensor network. It also considered hostile environment that adversaries can forge aggregation results. They combined a hop-by-hop data aggregation and end-to-end data secrecy. These two researches also motivated our work, since they considered compromised environments. Nonetheless, they are inappropriate for satisfying the smart grid requirements as mentioned in Section III.

In the Internet, capability-based DDoS defenses have presented to reserve network bandwidth for legitimate users. SIFF was presented by Abraham Yaar *et al.* [13], in which an end-host selectively stops individual flows. It classifies network traffic into eigher privileged and unprivileged packets, to guarantee bandwidth for privileged traffic from DDoS attacks. TVA by Yang *et al.* [14] proposed improved scheme against strategic attacks, such as flooding the setup channel, and exhausting the router state. However, these schemes cannot be naively adopted in the smart grid environment, since they were designed for the current Internet. In smart

grids, a pre-paid metering scheme [33] can be a capability-based approach, but none of them addresses privacy and security issues.

IX. Conclusion

In this paper, we propose secure and efficient capability-based power management to address reliablity, efficiency, privacy and security issues. It leverages a capability-based approach and secure data aggregation. The capability-based approach guarantees power resources for authorized users, and secure data aggregation using an asymmetric homomorphic encryption preserves private information and enables efficient data communications. Based on our numeric measurements, the proposed mechanism takes 11.12 seconds from aggregation phase to distribution phase, which is a affordable time for power providers. For future works, we plan to improve the proposed mechanism to address the open problems, such as instant detection for excessive capabilities.

Acknowledgment

This research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development, and Seoul R&BD Program(WR080951). Additionally, this research was supported by a grant NGIT2009100109 from the Northrop Grumman Information Technology Inc Cybersecurity Consortium.

References

[1] Korea Smart Grid Institute, "Korea's Jeju smart grid test-bed overview," Jun. 2009, http://www.smartgrid.or.kr/10eng3-1.php.

[2] J. Lu, D. Xie, and Q. Ai, "Research on smart grid in China," in *IEEE Transmission & Distribution Conference and Exposition: Asia & Pacific*, 2009, pp. 1–4.

[3] U.S. Department of Energy, "DOE Smart Grid System Report," Jul. 2009, http://www.oe.energy.gov/1446.htm.

[4] D. Novosel, M. M. Begovic, and V. Madani, "Shedding light on blackouts," *IEEE Power and Energy Magazine*, vol. 2, no. 1, pp. 32–43, 2004.

[5] National Institute of Standards and Technology (NIST), "NIST IR 7628: Guidelines for Smart Grid Cyber Security," Aug. 2010, http://csrc.nist.gov/publications/PubsNISTIRs.html.

[6] NERC, "Reliability Considerations from the Integration of Smart Grid," North American Electric Reliability Corporation, Tech. Rep., Dec. 2010.

[7] Federal Energy Regulatory Commission (FERC), "FERC's Smart Grid Policy," Jul. 2009, http://www.ferc.gov/industries/electric/indus-act/smart-grid.asp.

[8] A. D. Schmidt, H. G. Schmidt, L. Batyuk, J. H. Clausen, S. A. Camtepe, S. Albayrak, and C. Yildizli, "Smartphone malware evolution revisited: Android next target?" in *Proceedings of the 4th IEEE International Conference on Malicious and Unwanted Software (Malware 2009)*, 2009, pp. 1–7.

[9] Symmantec, "W32.Stuxnet," Sep. 2010, http://www.symantec.com/security_response/writeup.jsp?docid=2010-071400-3123-99.

[10] K. Hamilton and N. Gulhar, "Taking demand response to the next level," *IEEE Power and Energy Magazine*, vol. 8, no. 3, pp. 60–65, 2010.

[11] Wikipedia, "Rolling blackout," Nov. 2010, http://en.wikipedia.org/wiki/Rolling_blackout.

[12] R. E. Brown, "Impact of Smart Grid on distribution system design," in *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–4.

[13] A. Yaar, A. Perrig, and D. X. Song, "SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks," in *IEEE Symposium on Security and Privacy*, 2004.

[14] X. Yang, D. Wetherall, and T. E. Anderson, "TVA: a DoS-limiting network architecture," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1267–1280, 2008.

[15] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing internet denial-of-service with capabilities," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 39–44, January 2004.

[16] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2010, pp. 327–332.

[17] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security Privacy Magazine*, vol. 7, no. 3, pp. 75–77, 2009.

[18] H. Chan, A. Perrig, and D. X. Song, "Secure hierarchical in-network aggregation in sensor networks," in *ACM Conference on Computer and Communications Security*, 2006, pp. 278–287.

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999, pp. 223–238.

[20] Wikipedia, "Paillier cryptosystem," Aug. 2010, http://en.wikipedia.org/wiki/Paillier_cryptosystem.

[21] C. Wong, S. Bielski, A. Studer, and C. Wang, "Empirical Analysis of Rate Limiting Mechanisms," in *Proc. of the 11th International Symposium on Recent Advances in Intrusion Detection*, 2006.

[22] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling High Bandwidth Aggregates in the Network," *ACM SIGCOMM Computer Communications Review*, vol. 32, no. 3, pp. 62–73, Jul. 2006.

[23] NERC, "High-Impact, Low-Frequency Event Risk to the North American Bulk Power System," North American Electric Reliability Corporation, Tech. Rep., Jun. 2010.

[24] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-Grid Security Issues," *IEEE Security and Privacy*, vol. 8, pp. 81–85, 2010.

[25] Y. Liu, M. K. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," in *ACM Conference on Computer and Communications Security*, 2009, pp. 21–32.

[26] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2010, pp. 220–225.

[27] Crypto++, "Crypto++ 5.6.0 Benchmarks," Mar. 2009, http://www.cryptopp.com/benchmarks.html.

[28] D. Bickson, D. Dolev, G. Bezman, and B. Pinkas, "Peer-to-peer secure multi-party numerical computation," *Peer-to-Peer Computing, IEEE International Conference on*, vol. 0, pp. 257–266, 2008.

[29] A. Bartoli, J. Hernandez-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, "Secure information aggregation for smart grids using homomorphic encryption," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2010, pp. 333–338.

[30] A. Acquisti, "Receipt-free homomorphic elections and write-in ballots," CMU-ISRI-04-116, Carnegie Mellon University, Tech. Rep., 2004.

[31] K. Peng, R. Aditya, C. Boyd, and B. Lee, "Multiplicative homomorphic e-voting," in *In Advances in Cryptology - Indocrypt 04*, 2004, pp. 61–72.

[32] E. Mlaih and S. A. Aly, "Secure hop-by-hop aggregation of end-to-end concealed data in wireless sensor networks," *CoRR*, vol. abs/0803.3448, 2008.

[33] R. H. Khan, T. F. Aditi, V. Sreeram, and H. H. C. Iu, "A prepaid smart metering scheme based on wimax prepaid accounting model," *Smart Grid and Renewable Energy*, vol. 1, pp. 63–69, 2010.