International Journal of Parallel, Emergent and Distributed Systems, 2013
http://dx.doi.org/10.1080/17445760.2013.851677

No peeking: privacy-preserving demand response system in smart grids

Depeng Li[a]*, Zeyar Aung[b]*, John R. Williams[c] and Abel Sanchez[c]

[a]*Department of Information and Computer Science, University of Hawaii at Manoa, Honolulu, HI 96822, USA;* [b]*Computing and Information Science, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates;* [c]*Department of Civil and Environmental Engineering, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA*

(*Received 8 August 2013; accepted 27 September 2013*)

Demand response (DR) programs are widely used to balance the supply and demand of electricity in a smart grid. This results in a reliable electric power system. Unfortunately, privacy violation becomes a pressing challenge that drastically affects the DR programs. Power usage and operational data can be abused to infer personal information of customers. Without a well-designed privacy preservation mechanism, adversaries can capture, model and divulge customers' behaviour and activities. In this paper, we first investigate the natures of privacy leakages and explore potential privacy threat models. After that, we design and implement a new protocol named privacy-preserving demand response based on the attributed-based encryption, and formally prove its validity. To demonstrate its viability, the protocol is adopted in several types of DR programs on an emulated smart grid platform. Experimental results show substantially lighter overheads while formidable privacy challenges are addressed.

Keywords: data encryption; multicast; privacy; smart grids

1. Introduction

Smart grids facilitate smart energy management through active deployments of smart metering infrastructure in our society as part of a global initiative. An important feature in smart grid systems is the demand response (DR) program. In DR, customers alter their consumption patterns reacting to electricity price changes or utilities turn off customers' appliance when the system in jeopardy is sensed [30]. Through shaving power consumption peaks, DR lessens costs: even a slight power consumption decreases, e.g. 5% introduces significant, say 50%, cost-cutting. The reason is because electricity generation cost raises disproportionately when the power generation capacity is near its maximum limit [2].

1.1 Motivations

Despite aforementioned benefits, privacy leakages in DR have been widely discovered. DR together with smart metering technologies generates high-resolution data leaving customers' digital trails that others can monitor and exploit for their advantages. Without proper controls that help eliminate privacy violation, customers participating in DR face unpleasant experiences: loss of their sensitive information and disclosure of their activity patterns [19,22]. Thus, appropriate countermeasures are required to veil customers' privacy in a DR system.

*Corresponding authors. Email: depengli@hawaii.edu; zaung@masdar.ac.ae

Pioneer studies [16,32] realise that the privacy violation can occur due to the free access to power consumption data. To safeguard privacy, recent researches deploy batteries [19], cryptographic means [22, 23], etc., to hide/encrypt metering data at the smart meter end. However, privacy can be disclosed from other sources beyond power consumption data. In detail, DR systems contain several kinds of data: metering data, control commands, events, alarms, etc. [22]. The existing countermeasures mainly focus on the protection of electricity usages data. It is possible that the protection mechanisms could be bypassed while the adversary aims at other kinds of data. Messages sent from the utility to customers, for example, may trigger certain customers' reactions and in turn influence their power usage patterns. With the free access of those messages and contextual clues, scientific, curious or malicious users can not only infer customers' activity, but also deeply mine their habits such as their financial rationality. For example, at peak times, the electricity price is expensive. During that peak time, if a customer choose to turn off the air conditioner or raise the thermostat settings even though the outside temperature is baking hot, these particular behaviours can be mined to deduce that the customer prefers financial savings to the comfortably cool living temperature.

1.2 Our contributions

1.2.1 Privacy leakages

To our best knowledge, this paper is the first to study privacy leakages of DR programs. From an adversary's perspective, we practically illustrate privacy threats with the aid of corresponding examples. We further formalise two privacy leakage models, the benefit inconvenience evaluation (BIE) and the rationality inconvenience ratio (RIR). In BIE, the financial benefit resulting from rescheduling power consumption tasks is compared with the inconvenience that customers suffer. RIR compares customers' rationality with their discomfort experiences at every time instance. Thus, customers' cost saving against discomfort can be profiled. Such knowledge can be used in designing targeted advertisements for specific customers.

1.2.2 P2DR protocol

In this paper, we focus on privacy preservation in the DR program's communication system rather than only at the smart meter end. We develop a new fine-grained protocol named the privacy-preserving demand response (P2DR) protocol through the usage of the ciphertext-based attribute-based encryption (CP-ABE, in short, ABE) system [4]. This protocol is compatible with the existing DR model, which is managed and operated by the easily combined policy system based on residence addresses. It also offers high performance, and gives DR program a chance to take full advantage of ABE's flexibility. We further demonstrate how P2DR is utilised in a few real-world popular DR programs as examples.

1.2.3 Experimental validation on emulated smart grid platform

Finally, we implement our approach that is executed on the commodity control server and the emulated smart meters. The experimental results demonstrate that our solution merely incurs a low delay ($\leq$ 500 ms for number of attributes $<$15) which is acceptable to DR systems. Computational cost of P2DR is lightweight so that even emulated smart meters

that are configured with low-end CPU and limited memory in our experiment exhibit efficient performance.

2. Background

2.1 Demand response model

In smart grids, multicast is widely deployed due to its scalability, its efficiency and its functionality across network segments [31,34]. DR also uses it for the sake of efficiency. As depicted in Figure 1, the control server cooperates with smart meters to establish the DR protocol. Note that, in DR systems, a set of policies is established to manage the power curtailment. In this paper, we use *streets*, *ZIP*, *cities*, etc., as examples; some utilities may use '*district #*', '*sub-district #*', '*substation #*', '*feeder #*', etc. They are interchangeable in this paper since policies have to be translated into command messages before they are sent out to smart devices via multicast technologies.

2.2 Demand response program

The DR program aims to balance the supply and the load in real time. It can be classified into two categories [6]: (1) the time-based program such as *Day Ahead Pricing* (DAP) and (2) the incentive-based program such as *Direct Load Control* (DLC) and *Emergence Demand Response Program* (EDRP). For the former, customers can adapt their power usage regarding electricity price changes over times. The latter enables utilities to offer an incentive price encouraging customers to reduce their power consumption.

2.2.1 Direct Load Control program

While the stress status in supply, the DLC program enables the utility to remotely curtail customers' load in a short notice based on customers' prior consent [1]. The utility controls residents' smart appliances, for example, turning off air conditioners, water heaters and so on or changing their thermostat settings.

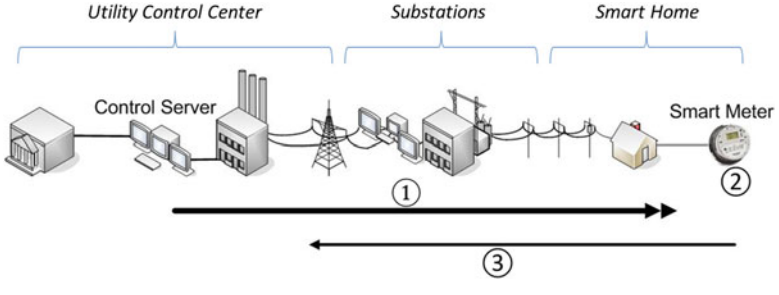

Figure 1. Demand response model in AMI. (1) A DR control server multicasts initial DR signals and subsequent DR events: $M = \{C_1\} \ldots \{C_x\}$ where $|M| = x$; (2) After receiving M, each smart meter sm_i decides if C_i is designated to itself. If so, DR events are incorporated into residences' energy plan or the command C_i is executed on its appliance; (3) After the execution of C_i, smart meters sm_i estimate and validate actual load shedding, confirm actual results and unicast them back to the DR control server if necessary.

2.2.2 Emergency Demand Response program

To mitigate the power demand during the peak times, utilities execute EDRP [1], one example of which is to solicit the customer enrolling in pre-defined promotion packages. Each package contains a time slot and an associated incentive price. After subscribing to a package, customers allow utilities to control their appliances in this particular time slot to get financial incentive in return.

2.2.3 Day Ahead Pricing

A day is usually divided into a number of time slots, for instance 24 one-hour slots, each of which is associated with a particular price. The DAP program [1] will inform the customers about the set of slots and corresponding prices in one-day ahead.

2.3 Group key, pairwise key and ABE

To safeguard messages in transmission, we discuss and compare different multicast encryption schemes or other cryptographic primitives. We argue that ABE appropriately satisfies flexibility requirements and efficiently accommodates DR's policies.

2.3.1 Group key scheme

To secure multicast communications, a symmetric group key (e.g. [15]) is used to encrypt/decrypt packets delivered among group members (e.g. smart devices). However, it results in expensive cost: for each policy/command, the Key Distribution Center (KDC) is used to enumerate all related smart devices and then to create a particular group; after that, partial keys/key encrypted keys (KEKs) are distributed to the group via secure channels; finally, each smart device in the group calculates the group key. Every new policy/command requires such a complicated procedure. The established group key is merely reusable. Therefore, we argue that there are substantial barriers to fully realise the group key scheme in collaboration with DR in smart grids.

2.3.2 Pairwise key

The pairwise key can be established between the key server and each smart device in DR. Its rate of data throughput is high and its key length is relatively short. However, it is mainly used in unicast communications but lacks the scalability for multicast. Furthermore, it raises complicated key management issues: (1) A number of key pairs should be managed in complicated smart grids. It results in the mandatory deployment of an unconditionally trusted third party. (2) The frequency to refresh session keys is high – the worst case is that each communication session demands a new session key [20]. Hence, it cannot be deployed in DR due to its expensive cost for key management and its inflexibility for multicast communications.

2.3.3 ABE

An ABE system [4] efficiently encrypts multicast messages in a fine-granular way. A control server encrypts each message based on corresponding policies associated with the message. A smart meter can decrypt ciphertext only if its associated attributes match with the policy of the ciphertext. Unlike other schemes that have to offer smart meters the private keys (e.g. the pairwise key or the group key) due to their coarse-grained

characteristics, ABE is fine grained and it can establish a specific access control policy on which smart meters can decrypt data. This satisfies the multicast service and the policy mechanism deployed in DR. However, ABE schemes require properly designed attribute sets in which the least number of attributes is used. Also, it demonstrates expensive computational overhead and communication cost.

3. Privacy leakage

In this paper, we realise that the direct access to utility control messages and metering data in DR offers a substantial potential for adversaries to easily infer customers' behaviour model, daily activities, habits, etc. [16]. We further present (1) general privacy leakage, (2) BIE model based on task rescheduling and (3) RIR model based on price changes.

Utilising the BIE and RIR models, an adversary is able to quantify customers' financial rationality against discomfort and then launch the following privacy violations: (1) Targeted advertisement: customers can be classified into *spendthrift, moderates* and *saver* types. Spendthrifts prefer to big-ticket items or even luxuries. Savers favour economical-and-applicable issues. Moderates are in between. Plus, based on habits to postpone tasks or bring tasks forward, customers are labelled as *early bird* or *latecomer* types. The former prefers to flyers beforehand and the latter enjoys a last-minute deal. Targeted advertisements could be customised for each one of them. (2) Alteration of customer types: when customers' type is altered, it may infer something new: there maybe new tenants or the resident may confront economic status changes.

In this section, based on the two models, we analyse two real-world scenarios: rescheduling cloth washing tasks and turning up/down the air conditioner. According to the result, the adversary could evaluate the level of inconvenience that customers can tolerate for the sake of financial gain.

3.1 General privacy leakage in DR system

Privacy threats occur when an adversary associates customers' fine-granular power usage data with daily activities, e.g. breakfast, laundry, wake-up cycles and so on [16,26,28]. Unlike previous research, we further observe that privacy violations also occur when adversaries infer from utility messages in a context of environmental information.

3.1.1 Privacy leakage via appliance malfunction

In a DLC program, utilities send a control command C_i to an appliance A_i which then executes C_i. When the execution fails and the appliance status S is sent back in a clear text, *Eve* is powerfully sufficient to capture the packet and to identify appliance malfunctions via analysing the status S. *Eve* then sells information to advertisement companies/ appliance manufacturers which send customers the targeted advertisements for the repairing or purchasing purpose.

3.1.2 Privacy leakage by customers' presence

A customer who enrols in an EDRP selects a specific package corresponding to a time slot, for example 13:00 to 15:00 to curtail the power usage since he/she is absent at that time. For example, the utility sends a remote control command to a participant ('*address A*') whose package corresponds to the time slot from 13:00 to 15:00. The command shuts

down the air conditioner though the temperature is high (e.g. >104°F/40°C). *Eve* can infer that residents are absent from 13:00 to 15:00 and he/she can take the risk to break in.

3.1.3 Privacy leakage by customers' financial incentives

Assume that a customer enrols in a DAP program. *Eve* can deduce the following habits of a customer: (1) Though being informed the varied prices, a customer does not reschedule tasks. It means that customers may not care about shaving off their electric bill. (2) In a building, if a customer reschedules his/her cloth washing task from daytime to midnight, it may infer that the customer pays little attention to his/her neighbours' reaction if the washing machine is noisy. (3) When electricity prices are expensive, the customer turns down the air conditioner even if it is a torrid day (e.g. the temperature >104°F/40°C). It means that the customer can tolerate discomfort for the sake of financial gain.

3.2 BIE model

The non-intrusive load monitoring (NILM) technology can be used to break the electricity demand profiles into different appliance usage tasks [4,25]. Through analysing those tasks, we propose a BIE model to evaluate the level of which a customer prefers financial gains while suffering conveniences in a DR program. Note that, for different appliances, the formula to evaluate the models can be different.

We assume that the adversary can collect two sets of DR messages, one before participating a DR program and the other after. The former is defined as m and the latter m'. Assume the adversary captures the demand profile in a time window with size x. To simplify, we assume that the time window is one day and the time step of each monitored time window is 1 h. So, $x = 24$. We define our privacy invasion method in the following: $N = \{\alpha_1, \ldots, \alpha_n\}$ is a set of appliances, where $|N| = n$; $\Psi = \{\psi_1, \ldots, \psi_m\}$ is a set of tasks, where $|\Psi| = m$; $M : N \times \Psi \rightarrow 0$ or 1 is a map between an appliance and a task; $D = \{d_1, \ldots, d_x\}$, where $|D| = x$ is a set of power demand/consumption and $P = \{p_1, \ldots, p_x\}$, where $|P| = x$ is a set of prices. According to the nature of the DR program, we also assume that $\forall\, d_i \in D$, there is one and only one corresponding $p_i \in P$.

Each task ψ_j (where $1 \leq j \leq m$) happening in the residence corresponds to an appliance α_i (where $1 \leq i \leq n$). The appliance can be a washing machine, an air conditioner and so on. A task ψ_i is defined as follows:

$$\psi_j = \left(S_j, F_j, L_j = F_j - S_j, \{d_{s_j}, \ldots d_{F_j}\}, \{p_{s_j}, \ldots p_{F_j}\}\right), \tag{1}$$

where $L_j > 0$. Note that the schedule of a task ψ_j starts at S_j, ends at F_j and takes $L_j = F_j - S_j$ time units. For each time unit from S_j to $S_j + L_j$, a task ψ_j demands d_x electricity and its corresponding price is p_x. Its costs C_{ψ_j} and its finance gain ΔC_{ψ_j} are listed in (2) and (3), respectively.

$$C_{\psi_j} = \sum_{t=S_j}^{F_j} d_t \times p_t, \tag{2}$$

$$\Delta C_{\psi_j} = C'_{\psi_j} - C_{\psi_j} = \sum_{t=S'_j}^{F'_j} d'_t \times p'_t - \sum_{t=S_j}^{F_j} d_t \times p_t. \tag{3}$$

To demonstrate BIE, two appliances, *washing machine* and *air conditioner*, are taken as examples:

3.2.1 Benefit versus inconvenience – cloth washing task

A customer reschedules a cloth washing task since the price at original time slots $[S_j, F_j]$ is more expensive than that in $[S'_j, F'_j]$. In formula (4), ΔC_{ψ_j} is the financial gain introduced from the rescheduling; $\Delta S_j = |S'_j - S_j|$ are the inconveniences that the customer has to suffer which is defined as the difference between the original cloth washing task ψ_j and the rescheduled task, ψ'_j. BIE for washing machines is defined as:

$$\mathrm{BIE}_{\mathrm{wash}} = \frac{\Delta C_{\psi_j}}{\Delta S_j} = \frac{\left(\sum_{t=S'_j}^{F'_j} d'_t \times p'_t - \sum_{t=S_j}^{F_j} d_t \times p_t\right)}{|S'_j - S_j|}. \tag{4}$$

3.2.2 Benefit versus inconvenience – air conditioner

For time slots $\{S_j, S_j + 1, \ldots, F_j\}$, we assume that $^\circ C = \{t_{S_j}, \ldots, t_{F_j}\}$ is the corresponding temperature that a customer set for an air conditioner before the DR and $^\circ C' = \{t'_{S_j}, \ldots, t'_{F_j}\}$ thereafter. ΔC_{ψ_j} is the financial gain and $^\circ C' - {}^\circ C$, the temperature difference, is the inconvenience that the customer suffers. BIE for air conditioners is defined as:

$$\mathrm{BIE}_{\mathrm{AC}} = \frac{\Delta C_{\psi_j}}{^\circ C' - {}^\circ C} = \frac{\left(\sum_{t=S_j}^{F_j} d'_t \times p'_t - \sum_{t=S_j}^{F_j} d_t \times p_t\right)}{\sum_{t=S_j}^{F_j} (t'_t - t_t)}. \tag{5}$$

3.3 RIR model

The price elasticity matrix (PEM) [4] is defined to assess customers' financial rationality when they participate in DR programs. But previous researches do not study the inconvenience a customer has to suffer in exchange for financial gain. Here, we describe quantified evaluation for RIR based on PEM.

The notion of *inconvenience* means the discomfort a customer experiences. It varies for different appliances. Here, we use an air conditioner's temperature as examples: customers set different temperatures to get financial gains based on the electricity prices at different time slots. Thus, *inconvenience* (namely, *ic*) is defined as:

$$ic_{t_i} = \frac{\partial t_{t_i}/t_0}{\partial p_{t_i}/p_0}, \tag{6}$$

where t_0 is the initial temperature, p_0 is the initial price, t_{t_i} is the temperature at time slot t_i and p_{t_i} is the electricity price at time slot t_i. RIR for the air conditioner is defined as:

$$\mathrm{RIR}_{\mathrm{AC}} = \frac{\sum_{i=1}^{24}\sum_{j=1}^{24} e_{i,j}}{\sum_{i=1}^{24} i \cdot t_{t_i}} = \frac{\sum_{i=1}^{24}\sum_{j=1}^{24} e_{i,j}}{\sum_{i=1}^{24} \left((\partial t_{t_i}/t_0)/(\partial p_{t_i}/p_0)\right)}, \tag{7}$$

where $e_{t,t'} = \partial d_t / \partial p_{t'}\ \forall t, t'$; t and t' are defined as different time instances; ∂d_t and ∂p_t are named as changes in demand and price at t and t', respectively. $e_{t,t}$ is referred to as cross self-elasticity. It indicates the change in demand at a time instance t due to the price change at the same time instance t. $e_{t,t'}$ is the cross elasticity. It is the change in demand at a time instance t due to the price change at the time instance t'. For $E = [e_{i,j}]_{24\times 24}$, $e_{i,x} = \partial d(t_i)/\partial p(t_x)$, where $x \in \{i, j\}$.

To get the financial gain, a customer can postpone tasks. The rationality to set tasks back is defined by adding up elements below diagonal in $E = [e_{i,j}]_{24\times24}$.

$$R_{\text{post}} = \sum_{i=1}^{24} \sum_{j=1}^{i-1} e_{i,j}. \tag{8}$$

To get financial gain, a customer can schedule tasks in advance. The rationality to fulfil tasks ahead of schedule is defined by adding up elements above diagonal in E.

$$R_{\text{prior}} = \sum_{i=1}^{24} \sum_{j=i+1}^{24} e_{i,j}. \tag{9}$$

4. Security overview and protocol

To protect the smart grid data against privacy leakages introduced in Section 3, we propose a protocol to secure the network communication.

4.1 Adversary model, security assumption and scope

4.1.1 Adversary model

Like other researches in areas of privacy preservation [8,13,22], we follow the semi-honest adversary model in which smart devices (e.g. smart meters and so on) obey DR. Meanwhile, they are also curious about messages they learn (or share). They have the intention to combine information and study sensitive messages if possible in order to uncover others' privacy.

4.1.2 Security assumption

We assume that smart devices such as smart meters are tamper resistant and device attestations are deployed to validate them. Furthermore, we also assume that the utilities deploy the PKI and the trusted KDC [13, 21]. To avoid the vulnerability that the confidentiality critically depends on the security of a single trusted KDC, CP-ABE can be replaced with the multi-authority CP-ABE [10, 24]. It will be our future work. So, this paper assumes that the KDC is trustworthy. Likewise, we assume that the control server and smart meters hide their own private keys and publish their public keys (e.g. RSA [20]).

4.1.3 Scope

Our protocol focuses on the confidentiality service to protect privacy between smart meters and utilities. Other security properties such as integrity and authentication services are also important but beyond this paper's scope. Insider attacks are not considered in this paper but will be our future research. The privacy protection between smart meters and appliances [11] is also out of the scope of this research.

4.2 System overview

There are three participants in the P2DR system: smart meters installed at customer residences as well as control servers and the trusted KDC deployed in the utility control centre.

There are two kinds of communications in a DR system: multicast and unicast. To protect the multicast communication that sends crucial DR messages from the control

server to multiple smart meters, we adopt an ABE encryption system [4] (for details, refer to Appendix B). To secure the unicast communication that sends results back from smart meters to the control server, we deploy the RSA public key encryption [20] which is faster as compared with others, e.g. El Gamal [20] in terms of encryption operations. It is important for smart meters that have the limited processing capability. The KDC's responsibility is to issue ABE keys and RSA private/public key pairs to smart meters and control servers.

A detailed view about how our P2DR adopts the ABE scheme is illustrated in Figure 2: at the set-up ❶ phase, the trusted KDC computes the ABE public key, $PK = \{\mathbb{G}_0; g; h = g^{\beta}; f = g^{1/\beta}; e(g,g)^{\alpha}\}$ and the ABE master secret key, $MSK = (\beta, g^{\alpha})$ by invoking the $Alg.9 - ABESetUp$. The next step for every smart meter is to register ❷ its own attribute sets, $\mathbb{S}$ to the trusted KDC. For example, a smart meter provides its attributes: $\mathbb{S} = \{$*street number: 12345; street name: main street; ZIP: xyz; city: noname*$\}$. After a successful authorisation, the trusted KDC generates ❸ the smart meter's ABE secret key, $SK = \{D, \{D_i\}\}$ via executing the $Alg.10 - ABEKeyGen$. Thereafter, every smart meter is issued ❹ its own SK and public key, PK by the trusted KDC in a secure channel (e.g. physical touch or RSA encryption). The control server receives the public keys via secure channels. Then, the control server encrypts ❺ plaintext M with the public keys, PK and the policy T of M, with the utilisation of $Alg.11 - ABEEncrypt$. The control server multicasts ciphertext CT and the policy T to smart meters. Smart meters can decrypt ❻ the ciphertext CT by running $Alg.12 - ABEDecrypt$ if the reflecting attribute sets T match with its own attributes, $\mathbb{S}$. Furthermore, if a smart device departs, its secret key should be revoked ❼ by our ABE rekeying scheme. For detailed Algorithms 1–8, refer to Section 4.3, and for detailed Algorithms 9–12, refer to Appendix B.

4.3 Protocol

The essential goal of the P2DR protocol is to realise an efficient privacy preservation mechanism satisfying scalability and time-critical requirements (according to some utilities' regulations, broadcast ≤ 15 s and multicast ≤ 5 s [17]) of a DR program without any privacy exposures. The P2DR protocol is compatible with existing DR systems [11].

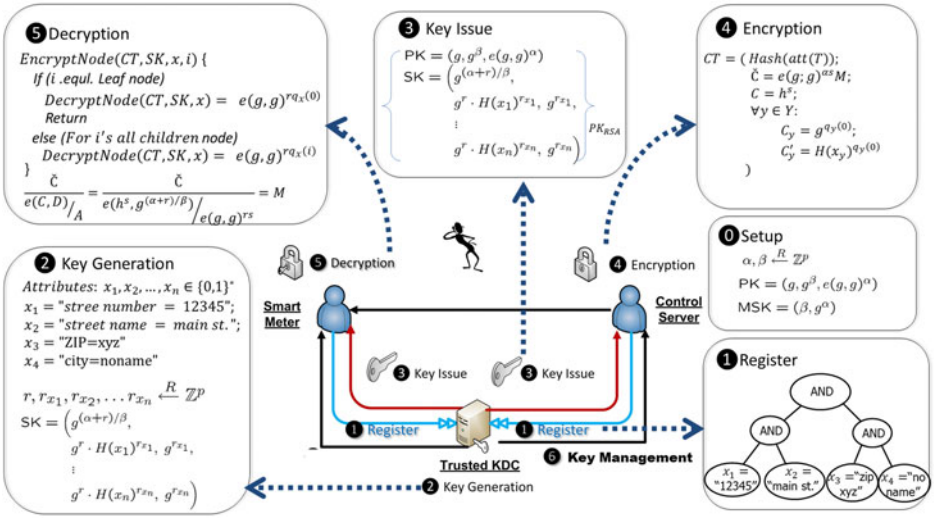

Figure 2. ABE system model.

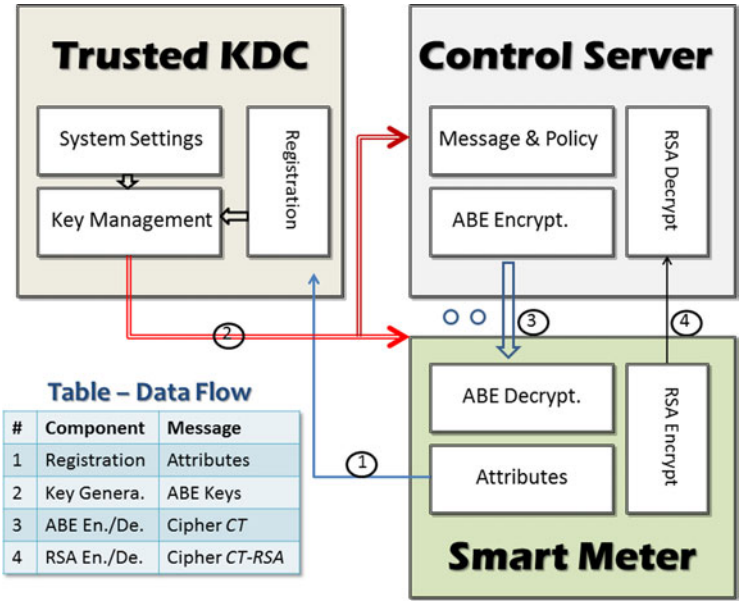

Figure 3. Architecture of P2DR protocol.

The basic components and their correlation of the P2DR protocol are depicted in Figure 3. How three DR programs, *DLC*, *EDRP* and *DAP*, incorporate with the P2DR protocol is described in this section as examples.

4.3.1 *System settings*

Step 1: During the set-up phase of the ABE scheme, a trusted KDC first creates the *PK* and *MSK* via invoking *Alg*. 1 −. *P2DRABESetup*. Thereafter, it publishes *PK* and hides *MSK*. Hence, the P2DR protocol takes advantage of the multicast encryption functionality and fine-grained attributes of the ABE scheme for the sake of efficiency.

Algorithm 1. P2DRABESet-Up(*PK, MSK, KDC*)

1: KDC End: Alg.9 ABESet-Up (*KDC.PK, KDC.MSK*)

(Note: Alg. 9 is in Appendix B.)

Step 2: The trusted KDC generates the RSA public and private key pair $(PB^{CS}_{RSA}, Pr^{CS}_{RSA})$ for the control server. PB^{CS}_{RSA} is published and Pr^{CS}_{RSA} is securely issued to the control server. For every smart meter, repeat this procedure.

4.3.2 *Registration*

The trusted KDC first assigns each smart meter an attribute set. Second, the attribute set is parsed into a standard attribute $\mathbb{S}$ via *Alg*. 2 *P2DRABEReg*. Later, $\mathbb{S}$ is used to create the smart meter's secret key (*SK*).

Algorithm 2. P2DRABEReg($\mathbb{S}$,*KDC, SMeter.Attributes*)

1: KDC End : $\mathbb{S} \leftarrow$ Parse (*SMeter.attribute)*
2: KDC.SMeter.$\mathbb{S} \leftarrow \mathbb{S}$
3: Smart Meter End: SMeter.$\mathbb{S} \leftarrow \mathbb{S}$

EXAMPLE I (DLC-P2DR). In DLC, we assume that there is a smart meter sm_i installed at each residence. A residential address is used as an attribute set to identify the smart meter sm_i. For example, for a smart meter *sm*, let $\mathbb{S}_{sm} = \{attr1 = `12345'; attr2 = `main\ street'; attr3 = `XYZ'; attr4 = `noname'\}$. In DLC-P2DR, the trusted KDC's registration component parses the attributes, $\mathbb{S}$.

EXAMPLE II (EDRP-P2DR). In EDRP, let us assume that a customer selects a promotion package corresponding to a time slot, for example from 13:00 to 15:00. In this time window, the control server can reduce the customer's power consumption through controlling customers' appliances. The following attributes, $\mathbb{S}_{sm} = \{attr1 \geq 13{:}00; attr2 \leq 15{:}00\}$, are assigned to a particular smart meter *sm*, in EDRP-P2DR.

EXAMPLE III (DAP-P2DR). In DAP, the control server will broadcast prices for 24-h time slots of the next day in advance. The information is plaintext which can be accessed by anyone.

4.3.3 ABE key generation

The trusted KDC first generates the secret key, *SK* for each smart meter via invoking $Alg.3 - P2DRABEGen$ with the input as smart meter's attributes $\mathbb{S}$ and ABE scheme's *MSK*. Then, the trusted KDC encrypts *SK* by using smart meter's RSA public key and deliver the ciphertext to *sm*.

Algorithm 3. P2DRABEGen(*KDC,SMeter.*$\mathbb{S}$, *SMeter.SK*)

1: KDC End: Alg.10 ABEKeyGen(*MSK, SMeter.*$\mathbb{S}$, *SK*)
(Note: Alg. 10 is in Appendix B.)
2: *cipher*← *RSAEnc(SK, SMeter.RSAPublicKey*)
3: Sends *cipher* to *SMeter*
4: Smart Meter End:
SMeter.SK ←*RSADec(cipher, SMeter.RSAPrivateKey)*

4.3.4 ABE encryption

DR messages such as prices, remote control commands, events, alarms and so on should be broadcasted in such a way that each smart meter can be informed. In P2DR, the control server first assigns a policy, $\mathbb{P}$ to reflect each message by invoking $Alg.4$.

Algorithm 4. P2DRAsgMsgPolicy(Message.$\mathbb{P}$, *Control Server, Policy*)

1: Control Server End: Message.$\mathbb{P}$← Parse (*Policy*)

The control server encrypts each message *M* with *M*'s policy, $\mathbb{P}$ and ABE's public key *PK* via invoking $Alg.5$. Then, the ciphertext *CT* and $\mathbb{P}$ are delivered to corresponding smart meters via multicast/broadcast channels.

Algorithm 5. P2DRABEEncry(*ControlServer,* $\mathbb{P}$, *M,PK,CT*)

1: Control Server End:
Alg.11 ABEEncryption($PK, M, \mathbb{P}, CT$) (Note: Alg. 11 is in Appendix B.)
2: Multicasts $\{\mathbb{P} \parallel CT\}$ *where* $\parallel$ is concatenation

Example I (DLC-P2DR). In order to achieve the demand and supply balance, the DLC delivers a command M to control air conditioners in a specific area. For example, the control server multicasts a control command M associated with the policy $\mathbb{P} = \{attr2 =$ '*main street*'; $attr3 =$ '*XYZ*'; $attr4 =$ '*noname*'$\}$. Our DLC-P2DR encrypts M with the ABE public key, PK and the policy $\mathbb{P}$. At last, the control server multicasts the ciphertext CT and $\mathbb{P}$ to smart meters in this area.

Example II (EDRP-P2DR). While peak times at a time slot, t, EDRP sends a command M to smart meters. If subscribing in a package which reflects the same time slot, smart meters execute M. For example, M's policy $\mathbb{P}$ is $\{attr1 \geq 13:00; attr2 \leq 15:00\}$. Our EDRP-P2DR encrypts the message M with the ABE public key, PK and $\mathbb{P}$. At last, the control server multicasts the ciphertext CT and $\mathbb{P}$ to smart meters. Only those enrolling for a particular package reflecting time slot [13:00,15:00] execute the command M and get the incentive payment in return.

4.3.5 *ABE key management*

We propose a centralised periodic batch rekeying scheme to manage the ABE keys regarding smart meters' joining or leaving. It provides both *backward secrecy* and *forward secrecy* [15]. *New joining* and *leaving* requests from smart meters are processed by the control server in a batch at the end of each rekeying interval. Our periodic batch rekeying strategy can alleviate the *out-of-sync* problem and improve the efficiency which is critical for smart meters with low-end device configurations. However, since the leaving smart meters can stay longer and the new meters have to join later, it introduces *vulnerability window* in which the security can be compromised. But smart meters are relatively static for a long time (e.g. a few years). In a short rekeying period (e.g. 1 day), the membership change events, e.g. *leaving* or *joining*, are rare. Thus, the security payback is tolerable and our periodic batch rekeying scheme is a trade-off between efficiency and security.

Unlike previous ABE encryption approaches (e.g. [4,24]), an ABE key scheme [9] based on the group key scheme presents *backward* and *forward* secrecy. However, for each attribute in the ABE system, a group key is required. Managing those group keys requests additional computational and communication overhead. Due to smart meters' limited resource, it may lead to the *out-of- sync* problem and thus cannot be used in smart grids directly. In our approach, we update the ABE keys with our periodic rekeying scheme and then encrypt the new ABE keys via RSA encryption scheme. The ciphertext is delivered in unicast. Detailed scheme is described below:

Step 1: Collect joining/leaving requests:

The trusted KDC collects *joining/leaving* requests from smart meters sm_i in the interval of a rekeying period, p:

$$sm_i \rightarrow KDC : R = \{Request \| S_i\}$$

where $1 \leq i \leq m$; $S_i = \{x_{i1} \ldots x_{in}\}$ is sm's attribute set

Step 2: Mark impacted smart meters

For smart meters which join/leave in a period, p, the trusted KDC enumerates all of sm_i's attributes S_i. which are stored in W, a matrix. W contains all impacted attributes that

need to be renewed:

$$W = \begin{pmatrix} S_1 \\ \vdots \\ \vdots \\ S_m \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1 \cdot m_1} \\ \vdots & \ddots & \vdots \\ x_{m_1} & \cdots & x_{m \cdot m_n} \end{pmatrix}, \tag{10}$$

where $\{\forall x_{ij} : x_{ij} \in S_i; \forall S_i : S_i \in R\}$.

Then, we mark each smart meter that has one or more than one attribute belonging to W. In details, for each smart meters sm and its attribute set, S_i, verify whether there is $x_{ij} \in W$ existing and $\forall x_{ij} \in S_i$. If so, mark the smart meter, sm as impacted.

Step 3: Rekey

At the end of the rekeying period, the trusted KDC will update a smart meter sm's secret key SK if sm is marked:

- Generate a random $r_{\text{new}} \overleftarrow{R} \mathbb{Z}_p$.
- Update ABE PK: $e(g,g)^{\alpha + r_{\text{new}}}$
- For each attribute $x_{ij} \in S_i$,

  if $x_{ij} \in W$, create a random $r_j^{\text{new}} \overleftarrow{R} \mathbb{Z}_p$
- Calculate

$$SK' = D = g^{\frac{\alpha + r_{\text{new}}}{\beta}}; \{\forall x_{ij} \in S_i : (D_j, D_j^{'}) = \begin{cases} D_j = g^{r_{\text{new}}} \times H(j)^{r_j^{\text{new}}}; D_j' = g^{r_j^{\text{new}}} \; x_{ij} \in W \\ D_j = g^{r_{\text{new}}} \times H(j)^{r_j}; D_j' = g^{r_j}, x_{ij} \notin W \end{cases} \tag{11}$$

Step 4: Deliver renewed/new ABE key

For marked or new joining smart meters sm_i, the trusted KDC encrypts the updated ABE secret key $SK^{'}$ with sm_i's RSA public key RSA_{sm_i} which is delivered to sm_i:

$$KDC \rightarrow sm_i : CT = \{SK'\}RSA_{sm_i}$$

4.3.6 ABE decryption

After successfully receiving ciphertext CT and $\mathbb{P}$, each smart meter sm first matches its own attributes $\mathbb{S}_{sm}$ with $\mathbb{P}$ to decide whether the ciphertext CT is designated to itself. If so, sm decrypts the ciphertext CT via invoking Alg.6. After then, sm can read message C_i and associates C_i into its energy use plan or executes C_i directly.

Algorithm 6. P2DRABEDecryption(*SmartMeter,SK,PK,CT*)

1: Smart Meter end: Match($\mathbb{S}_{sm}$, $\mathbb{P}$)
Alg.12 ABEDecryption(PK, SK, CT, M) (Note: Alg. 12 is in Appendix B.)

Example I (DLC-P2DR). In DLC, let smart meter *sm*'s attributes $\mathbb{S}_{sm} = \{attr1 = \text{'12345'}; attr2 = \text{'main street'}; attr3 = \text{'XYZ'}; attr4 = \text{'noname'}\}$. Since ciphertext CT's policy $\mathbb{P}$ is $\{attr2 = \text{'main street'}; attr3 = \text{'XYZ'}; attr4 = \text{'noname'}\}$, $\mathbb{S}_{sm}$ matches with $\mathbb{P}$. Therefore, the smart meter decrypts the ciphertext CT and gets the message M.

Example II (EDRP-P2DR). In EDRP, let smart meter *sm*'s attributes $\mathbb{S}_{sm} = \{attr \geq 13{:}00; attr2 \leq 15{:}00\}$. Since ciphertext CT's policy $\mathbb{P}$ is $\{attr1 \geq 13{:}00; attr2 \leq 15{:}00\}$, $\mathbb{P}$ matches with $\mathbb{S}_{sm}$. Therefore, the smart meter decrypts the ciphertext CT and gets the message M.

4.3.7 *RSA encryption*

Each smart meter needs to report its status, execution result, etc., to the control server so that the DR program can verify whether the real-time power usage complies with the balance principle. Therefore, in P2DR, the smart meter encrypts its result or its status via invoking *Alg*. 7 with the control server's RSA public key PB_{RSA}^{CS}. Then, the ciphertext CT-RSA is sent back to the control server which is the only one that can decrypt it.

Algorithm 7. P2DRRSAEncrypt(*RSAPubKCS, M, CT-RSA*)

1: Smart Meter end:
CT-RSA ← RSAEncryption(*M, RSAPubKCS*)

4.3.8 *RSA decryption*

After receiving ciphertext *CT-RSA* sent from the smart meter, *sm*, the control server invokes *Alg*. 8 with its RSA public key as input to decrypt it. The outputted plaintext M will be used by the control server to evaluate the accomplishment of the mission designated to *sm*.

Algorithm 8. P2DRRSADecryption(*RSAPriKCS,M, CT-RSA*)

1: Control Server end:
M ← RSADecryption(*CT-RSA, RSAPriKCS*)

5. Privacy evaluation

In this section, we critically examine our P2DR system based on the generic bilinear group model [3]. We argue that it meets the data privacy, namely distinguishability under the chosen-plaintext attack (CPA) and the adaptive chosen-ciphertext attacks (CCA) as no efficient adversary with any reasonable probability can break P2DR. Without the direct access to DR data, privacy attacks aforementioned, e.g. regular privacy leakages, RIR, BIE and so on, cannot be successful.

We know that if the ciphertext generated by the ABE scheme or the RSA public key system is probably secure, the ciphertext delivered on communication channels of the P2DR system can provide the data privacy. Thus, in this subsection, we prove that ABE and RSA components in the P2DR system are secured sufficiently.

We first describe the Decisional Bilinear Diffie–Hellman (D-BDH) assumption which is the cornerstone of the P2DR protocol's semantic security we are going to prove. Second,

we prove the security of ABE components utilised in P2DR. Third, we prove the security of the RSA public key encryption component in P2DR.

5.1 Data privacy in ABE component of P2DR

5.1.1 D-BDH assumption

Let $a, b, c, z \xleftarrow{R} \mathbb{Z}_P$. There are two tuples: $(A = g^a, B = g^b, C = g^c, \hat{e}(g,g)^{abc})$ and $(A = g^a, B = g^b, C = g^c, \hat{e}(g,g)^z)$. The D-BDH assumption is that no probabilistic polynomial time algorithm $\mathcal{A}$ can distinguish them with more than a negligible advantage. $\mathcal{A}$'s advantage is defined as:

$$Adv_{\mathcal{A}} = \left| \Pr\left[\mathcal{A}(A, B, C, \hat{e}(g,g)^{abc}) = 0\right] - \Pr\left[\mathcal{A}(A, B, C, \hat{e}(g,g)^{z}) = 0\right] \right|. \quad (12)$$

Definition 1 (ABE-CPA). Let $\mathcal{P} = (\mathcal{S}, \mathcal{G}, \mathcal{E}, \mathcal{D})$ be the ABE system in P2DR which encrypts/decrypts utility messages M in transmission. $\mathcal{S}$ stands for ABE set-up, $\mathcal{G}$ for ABE key generation, $\mathcal{E}$ for ABE encryption and $\mathcal{D}$ for ABE decryption. Let $b \in \{0, 1\}$. Let $\mathcal{A}$ denote an adversary which can access the ciphertext, CT.

We say that ABE-CPA holds the semantic security under CPAs launched by all polynomial time complexity adversaries $\mathcal{A}$ if $\mathcal{A}$'s $Adv_{\mathcal{P},\mathcal{A}}^{ABE-CPA-b}(k)$ is negligible. The security model we are going to use follows the experiment below:

Experiment $Exp_{\mathcal{P},\mathcal{A},\mathcal{A}}^{ABE-CPA-b}(k)$

$(PK, MSK) \xleftarrow{R} \mathcal{S}(k)$;
$SK \xleftarrow{R} \mathcal{G}(MSK)$;
$M_0 \xleftarrow{R} \{0,1\}^*$; $M_1 \xleftarrow{R} \{0,1\}^*$;
$CT_b \leftarrow \mathcal{E}(PK, M_b)$;
$M_b \leftarrow \mathcal{A}(\text{find}, CT_b, M_0, M_1)$;
return: $g \leftarrow \mathcal{A}(\text{guess}, CT_b)$

Briefly, there is a security game experiment with the parameter k where k is the bit length. An adversary $\mathcal{A}$ is given a set of public keys which can be used by $\mathcal{A}$ to generate any numbers of ciphertexts within polynomial bounds. The adversary $\mathcal{A}$ provides the challenger two messages M_0 and M_1. The challenger flips a fair coin $b \in \{0, 1\}$ and encrypts M_b. During the experiment, the adversary $\mathcal{A}$ can query for any private keys but is not allowed to use them for any decryption. At some time points, $\mathcal{A}$ outputs a guess bit $g \in \{0, 1\}$. We say that $\mathcal{A}$ wins the game if $g = b$ but fails otherwise. Based on the experiment, the adversary $\mathcal{A}$'s advantages can be defined as:

$$Adv_{\mathcal{P},\mathcal{A}}^{ABE-CPA-b}(k) = \Pr\left[Exp_{\mathcal{P},\mathcal{A}}^{ABE-CPA-0}(k) = 0\right] - \Pr\left[Exp_{\mathcal{P},\mathcal{A}}^{ABE-CPA-1}(k) = 0\right]. \quad (13)$$

$$= 2 \cdot \Pr\left[Exp_{\mathcal{P},\mathcal{A}}^{ABE-CPA-0}(k) = 0\right] - 1. \quad (14)$$

THEOREM 1. Suppose D-BDH assumption holds. There is no polynomial time adversary $\mathcal{A}$ that can break the semantic security of ABE components in P2DR system by CPA or CCA.

Proof. Suppose we have an adversary $\mathscr{A}$ with negligible advantage $\varepsilon = Adv_{\mathcal{P},\mathscr{A}}^{ABE-CPA-b}(\cdot)$ which can break ABE components in P2DR system. A simulator $\mathcal{B}$ which plays the decisional BDH game with the advantage ε processes in the following way:

Init Let the adversary $\mathscr{A}$ randomly choose the set of challenge access structures, namely T∗ which will be challenged upon.

Setup The simulator $\mathcal{B}$ first randomly generates two credentials, $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$. Then, $\mathcal{B}$ sends adversary $\mathscr{A}$ the following public keys:

$$PK = \left\{ \mathbb{G}_0;\ g;\ h = g^{\beta};\ f = g^{1/\beta};\ e(g,g)^{\alpha} \right\}$$

We then show how $\mathcal{B}$ programs each node $y \in Y$, where Y is a set of leaf nodes in the access structure T∗:

The simulator $\mathcal{B}$ calculates the following pair, $\{C_y = g^{q_y(0)};\ C_y' = H(att(y)^{q_y(0)}$, *where* $q_y(0)$ is based on $s \overset{R}{\leftarrow} \mathbb{Z}_p$. Note that the *att*() function returns attributes which can be any string $\in \{0,1\}^*$.

Phase 1: $\forall i$, a string, the adversary $\mathscr{A}$ evaluates $H(i)$ by randomly generating $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$. The simulator $\mathcal{B}$ provides g^{t_i} in response. For the set s_j of attributes, the adversary $\mathscr{A}$ makes the j'th key generation query. In response, the simulator $\mathcal{B}$ generates $r^{(j)} \overset{R}{\leftarrow} \mathbb{Z}_p$ and $\forall i \in s_j$, $r_i^{(j)} \overset{R}{\leftarrow} \mathbb{Z}_p$. Then, the simulator $\mathcal{B}$ calculates:

$$D = g^{(\alpha + r^{(j)})/\beta}; \quad \text{and} \quad \forall j \in s_j : \left\{ D_i = g^{r^{(j)} + t_i r_i^{(j)}}; D_i' = g^{r_i^{(j)}} \right\}$$

Then, they are sent to the adversary $\mathscr{A}$.

Challenge: the adversary $\mathscr{A}$ submits two challenge messages M_0 and M_1 and the access tree T∗ to the simulator $\mathcal{B}$. The simulator $\mathcal{B}$ needs to compute one of $M_0 \hat{e}(g,g)^{\alpha s}$and $M_1 \hat{e}(g,g)^{\alpha s}$, where $\alpha, s \overset{R}{\leftarrow} \mathbb{Z}_p$. Here, we consider a modified game where $\tilde{C}$ is calculated by either $\hat{e}(g,g)^{\alpha s}$ or $\hat{e}(g,g)^{\theta}$ where $\theta \overset{R}{\leftarrow} \mathbb{Z}_p$ Therefore, the adversary $\mathscr{A}$ with advantage ε for the ABE component in P2DR can be transformed into a new adversary with the advantage of $\varepsilon/2$. To simplify, we use the modified game from now on. Based on the notions aforementioned, the simulator $\mathcal{B}$ processes the followings: First, $s \overset{R}{\leftarrow} \mathbb{Z}_p$. Then, the linear secret sharing scheme associated with the access tree is used to construct the share λ_i of s for all relevant attributes, i. Third, the simulator $\mathcal{B}$ chooses $\theta \overset{R}{\leftarrow} \mathbb{Z}_p$. Fourth, the simulator $\mathcal{B}$ flips a fair coin $\mu \in \{0,1\}$ which is beyond the awareness of the adversary $\mathscr{A}$. At last, accomplish the following encryption:

$$\tilde{C} = M_{\mu} e(g,g)^{\theta}; \quad C = h^{s}; \quad \forall i \in Y : \{C_i = g^{\lambda_i}; C_i^{`} = g^{t_i \lambda_i}\};$$

They will be sent to adversary $\mathscr{A}$.

Phase 2: the simulator $\mathcal{B}$ repeats what it did in Phase 1.

Guess: the adversary $\mathscr{A}$ eventually submits a guess b of μ. If $b = \mu$, the simulator $\mathcal{B}$ will output 0 to note that $T = e\left(g,g\right)^{\theta}$. If $b \neq \mu$, the simulator $\mathcal{B}$ will output 1 which means that T is evaluated as a random group element of $\mathbb{G}_{\mathbb{T}}$. In the case that T is the expected element for which the simulator $\mathcal{B}$ provides a perfect simulation, we can deduce that:

$$Pr[B(PK, D, D_i, T = e(g,g)^{\theta}) = 0] = 1/2 + Adv_{\mathscr{A}}. \tag{15}$$

Otherwise, T is a random group element. It means that the adversary $\mathcal{A}$ cannot correctly decide which message M_μ is. Therefore, we have

$$Pr[B(PK, D, D_i, T = Random) = 0] = 1/2. \tag{16}$$

Consequently, the simulator $\mathcal{B}$ plays the decisional BDH game with non-negligible advantage.

In our ABE rekeying scheme, for each impacted attribute, its corresponding random value will be refreshed which is then used to update the ABE secret key of each impacted smart meter. Adversaries may try to launch the collision attack to take advantage of the fact that a set of smart meters is bound with this specific random value. The *phase 1*, *challenge*, *phase 2* and *guess* and the rest games can be repeated. Using the similar formal proof we mentioned earlier, we can conclude that the simulator $\mathcal{B}$ plays the decisional BDH game with non-negligible advantage. This can also be applied to the *new joining* smart meters.

So, it proves the CRA in the P2DR protocol by allowing random oracle techniques for decryption in *Phase 1* and *Phase 2*. This can also be extended to prove the CCA for decryption in *Phase 1* and *Phase 2*. □

5.2 Data privacy in RSA component of P2DR

5.2.1 DDH assumption

Let $a, b, y \xleftarrow{R} \mathbb{Z}_P$. There are two tuples: $(A = g^a, B = g^b, g^{ab})$ as well as $(A = g^a, B = g^b, g^c)$. The Decisional Diffie–Hellman (DDH) assumption is that no probabilistic polynomial time algorithm $\mathcal{B}$ can distinguish them with more than a negligible advantage. $\mathcal{B}$'s advantage is:

$$Adv_{\mathcal{B}} = \left| \Pr[\mathcal{B}(A, B, g^{ab}) = 0] - \Pr[\mathcal{B}(A, B, g^c) = 0] \right| \tag{17}$$

Definition 2 (RSA-CPA). Let $\mathcal{P} = (\mathcal{S}, \mathcal{G}, \mathcal{E}, \mathcal{D})$ be the RSA public key system in P2DR which encrypts/decrypts metering messages M in transmission from smart meters to the control server. $\mathcal{S}$ stands for RSA Set-up, $\mathcal{G}$ for RSA key Generation, $\mathcal{E}$ for RSA Encryption and $\mathcal{D}$ for RSA Decryption. Let $b \in \{0, 1\}$. Let $\mathcal{A}$ denote an adversary which can access the ciphertext, CT.

Theorem 2. Suppose the DDH assumption holds. There is no polynomial time adversary $\mathcal{A}$ that can break semantic security of RSA public key components in P2DR system by CPA or CRA.

Proof. The model utilised in Theorem 1 is easily reused to prove RSA-CPA and RSA-CRA in P2DR. □

6. Experiments and performance evaluation

The performance for the P2DR protocol depends on two critical parts: (1) *ABE Encryption* and *RSA Decryption* components at the control server end as well as (2) the *ABE Decryption* and *RSA Encryption* components at the smart meter end. As depicted in Figure 3, they correspond to the message transmission ③ and ④, respectively. Note that the

signal propagation delay is negligible. Times used by the *Message & Policy* component are trivial. We will not discuss their performance due to page limits.

6.1 Experiments

We implement ABE based on the pairing-based cryptography (PBC) library [18] built on the GNU multiple precision (GMP) arithmetic library (http://gmplib.org/): the GMP library provides arbitrary precision arithmetic APIs that are invoked by PBC to support the pairing-based cryptosystem. In our application, we use the pairing-friendly elliptic curves $E(\mathbb{F}_{2^{379}})$: $y^2 + y = x^3 + x + 1$ and $E(\mathbb{F}_p) : y^2 = x^3 + Ax + B$ with a 512-bit prime. Furthermore, to satisfy the performance requirement, we deploy the MNT elliptic curve to implement the ABE scheme. In Figure 4, we demonstrate those functions' performance when executing

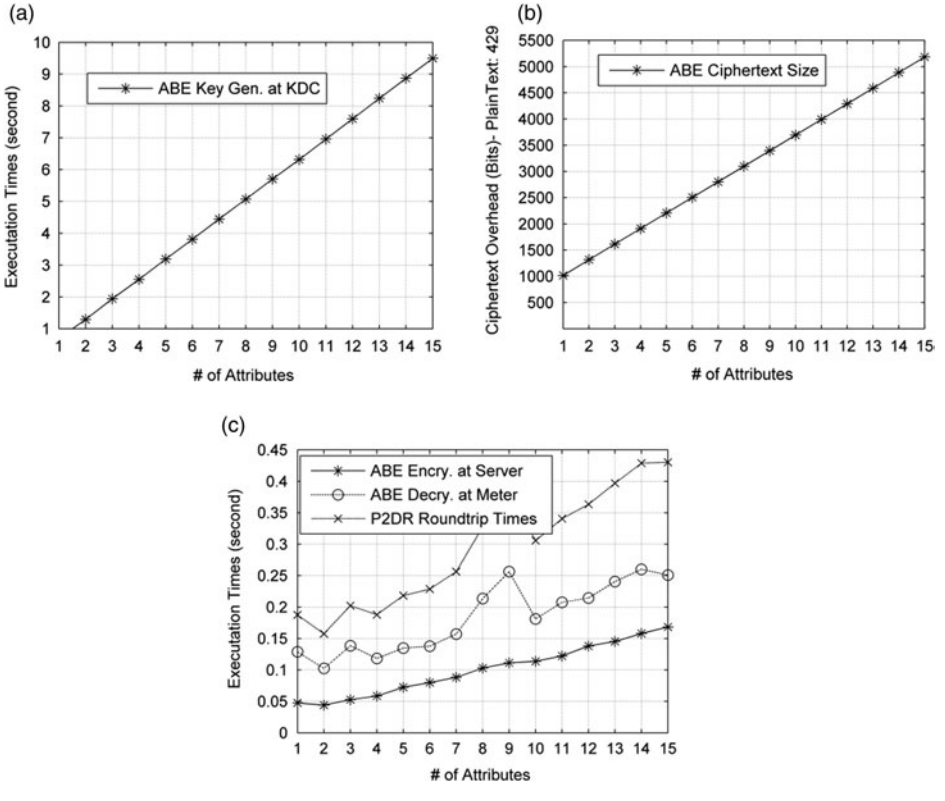

Figure 4. Test results of execution times. MNT elliptic curve of embedding degree 6 with order 160 bits length and base field order 512 bits length were utilised in P2DR. We collected 10 times' (randomly selected number) executions of ABE operations, the average values of which are illustrated in (a)–(c), including (a) ABE key generation on a KDC, (b) ABE Ciphertext size when the plaintext size is 429 bits and (c) the round-trip time of the P2DR protocol to process a command sent from the control server to smart meters and vice versa (the propagation delay is too trivial to be included). The number of attributes ranged from 1 to 15 (the maximum in practical case). As executing unauthorised third party system software upon real-world smart meters is prohibited (according to GE Company), the control server/KDC and the smart meter in the experiment were both virtual machines hosted by Oracle's VirtualBox installing Ubuntu 11.10. The detailed configuration of KDC/control server: memory – 4 GB; CPU – 2.67 GHz; Disc – 7.9 GB. That of the smart meter is memory – 4 MB; CPU – 33 MHz which is the same configuration of a typical real-world smart meter CPU.

them on a control server, a KDC and a simulated smart meter, respectively. Figure 4(a) demonstrates the execution times utilised to generate ABE keys on the KDC when the number of attributes ranges from 1 to 15.As illustrated in Figure 4(c), the ABE encryption at a control server and the ABE decryption at a simulated smart meter execute less than 170 ms and 260 ms, respectively, when the number of attributes is 15 or less. The round-trip execution time for P2DR takes less than 450 ms when the number of attributes is 15 or less. In words, the P2DR system can satisfy the DR program's requirement because the DR program accepts up to 5 seconds' delay in terms of the round-trip time [17].

The ABE communication overhead for P2DR is illustrated in Figure 4(b). It shows the bit sizes of ciphertext transmitted in P2DR with attribute numbers ranging from 1 to 15. Though communication overhead is still affordable in DR, its reduction is highly demanded. Note that the design of attributes in P2DR should minimise the number of attributes. We conclude that they are efficiently sufficient to be utilised in the DR system.

6.2 Evaluations

Table 1 evaluates the number of operations for each component in different ABE schemes. Performance of components in ABE is listed in Table 2. Previous ABE schemes [4,24], etc., which utilise one more attribute, namely the expiration date, to expire, validate or update the ABE key, demand two more exponentiations for each ABE operation. Based on our experiments, it takes extra time for the round trip of our P2DR protocol. An ABE rekeying scheme in [9] deploys the group key agreement for each ABE attribute to update each ABE key. Its computation cost including both the ABE rekeying and the group rekeying is listed below (Refer variables/symbols in footnote of Table 1):

$$F_{[24],\mathrm{ABE}} = (J+L)\left(\sum_{\forall sm \in S,\, sm=1}^{S|} |A_{\mathrm{sm}}|\right)E \tag{18}$$

$$F_{[24],\mathrm{GpKey}} = (J+L)\left(\sum_{t=1}^{|A|} d(\log_d T_t)\right)|S|\cdot C \tag{19}$$

Our ABE rekeying scheme including the ABE Rekeying and the RSA public key encryption and the RSA public key decryption is evaluated as follows:

$$F_{2,\mathrm{ABE}} = 2\left(\sum_{\forall sm \in S,\, sm=1}^{|S|} |A_{\mathrm{sm}}|\right)E \tag{20}$$

$$F_{2,\mathrm{RSA}} = 2\left(\sum_{\forall sm \in S,\, sm=1}^{|S|} |A_{\mathrm{sm}}|\right)E \tag{21}$$

After comparing with other solutions, we observe that our approach demonstrates the scalability and is more efficient. However, our scheme also shows vulnerability windows.

6.3 Possible improvements

In our scheme, a few open problems are left which are both intriguing and challenging. We will address them in our future research and meanwhile hope to inspire further efforts in a

Table 1. Performance evaluation: comparison of ABE components in P2DR with others.

Component		Cost of original ABE key scheme		Cost of ABE rekey via group key [9]		Cost of our P2DR	
		Computation	Communication	Computation	Communication	Computation	Communication
ABE key	Encryption	$2E(1+\|AT\|)$	$2(1+\|\text{AT}\|)\|g\|$	$2E\|\text{AT}\|$	$2\|\text{AT}\|\cdot\|g\|$	$2E\|AT\|$	$2\|\text{AT}\|\cdot\|g\|$
	Decryption	$\leq (2P(1+\|AT\|)+E\cdot H\cdot\|N_{sm}\|)$	$\|M\|$	$\leq (2P\cdot\|AT\|+E\cdot H\cdot\|N_{sm}\|)$	$\|M\|$	$\leq (2P\cdot\|AT\|+E\cdot H\cdot\|N_{sm}\|)$	$\|M\|$
ABE key update	Generation	$2E\cdot(\|A_{sm}\|+1)$	$2(1+\|A_{sm}\|)\|g\|$	$2E\cdot\|A_{sm}\|$	$2\|A_{sm}\|\cdot\|g\|$	$2E\cdot\|A_{sm}\|$	$2\|A_{sm}\|\cdot\|g\|$
	Update	$2(J+L)E\|A_{sm}\|$	$2(J+L)\cdot(1+\|A_{sm}\|)\|g\|$	$\leq (F_{[24],\text{ABE}}+F_{[24],\text{GpKey}})$	$2(J+L)\cdot(\|A_{sm}\|\cdot\|g\|+\sum_{t=1}^{\|A\|}\log_d T_t\|G\|\cdot\|S\|)$	$\leq (F_{\text{our,ABE}}+F_{\text{our,RSA}})$	$2\|A_{sm}\|\cdot\|g\|$

E, exponentiation; P, pairing; $g \in \mathbb{G}$; C, symmetric encryption; $|J|$, number of *new joining* sm; $|L|$, number of *leaving* sm; $|S|$, number of all sm; $|G|$, key length of group key G; sm, smart meter; $|A_{\text{sm}}|$, number of attributes given to sm; $|\text{AT}|$, number of Leaves in ciphertext Access Tree (AT); $|N_{\text{AT}}|$, number of nodes in ciphertext Access Tree (AT); $|\text{A}|$, number of attributes. In P2DR; d, degree of group key tree; $|T_t|$, number of sm associated with an attribute t; Refer $F_{[11],\text{ABE}}$,$F_{[11],\text{GpKey}}$, $F_{\text{our,ABE}}$,$F_{\text{our,RSA}}$ in Equations (17)–(20).

Table 2. Execution times of cryptographic components.

Item	Host	Time (ms)
ABE set-up	Trusted KDC	26.45
RSA encryption	Smart meter	8.096
RSA decryption	Control server	2.952

strong sense: (1) Insider attackers which own the same set of attributes and ABE keys can eavesdrop the sensitive messages and break the privacy; (2) Trusted KDCs could be compromised by malicious parties which can obtain all ABE keys issued to smart meters. Those ABE keys can be misused by adversaries to decrypt ciphertext in P2DR; (3) the multiple-authority ABE schemes are desirable to be adopted in P2DR.

7. Related works

Several means can extract privacy of power usage data in the DR program. Lisovich *et al.* [16] conduct a live monitoring experiment in a residence. They use NILM to extract and to analyse appliance usage based on collected power demand data with a time resolution of 15 s. They further design a behaviour extraction algorithm to measure critical privacy parameters (presence, sleep cycle, number of residence, etc.). In [32], Wicker and Thomas propose a framework guided by privacy-aware design practices (HEW methodology). Cho *et al.* [5] propose AERO to extract user's activities based on the Activities of Daily Living (ADL).

7.1 *Privacy preservation for smart metering*

Researchers also study means to preserve privacy for smart metering technologies but they are not designed specifically for DR. (**1**) ***Battery***: McLaughlin *et al.* [19] develop the Non-Intrusive Load Levelling (NILL) to hide the appliance's power usage signature via rechargeable batteries. Yang *et al.* [33] propose a stepping framework which outperforms other battery-based load hiding algorithms, e.g. NILL. However, rechargeable batteries are costly ($1,000 [19]) and labour-intensive. (**2**) ***Anonymity***: Efthymiou and Kalogridis [7] propose a trusted key escrow service to anonymise frequent readings with pseudonymous IDs for metering data. (**3**) ***Disturbance***: Li *et al.* [14] design a privacy-enhancing approach via compressing meter readings. Tomosada *et al.* [29] propose a method to generate virtual demand data which can be distributed among institutes to protect customer's privacy. (**4**) ***Cryptographic Schemes***: Li *et al.* [13] protect smart metering data aggregation via the homomorphic encryption algorithm. Garcia and Jacobs [8] design a privacy-friendly protocol by using homomorphic (Paillier) encryption and additive secret sharing. Rial and Danezis [27] use zero knowledge proofs and commitments to preserve smart meters' privacy. Li *et al.* [12] proposed the privacy preservation scheme that mainly focused on remote appliance control programs. It showed the limit that the number of attributes is 5 or less which may not satisfy the real smart grid system. Furthermore, the operational results are sent back from smart meters to the utilities in the plaintext format, and it can possibly be a privacy leakage.

7.2 *ABE revocation schemes*

The original ABE system [4] or its variants [24] revoke ABE keys via expiration dates. Another ABE key revocation scheme [24] uses negative clauses. However, (1) their

performances are not efficient for smart meters since they add one more attribute for each ABE encryption/decryption. (2) They cannot efficiently process *new joining* smart meters with *backward secrecy*. The attribute revocation scheme [11] utilises the group key scheme (e.g. [15]) to generate attribute group keys which can encrypt the updated ABE secret keys. It satisfies the characteristics of Disruption-Tolerant Military Network – intermittent network connectivity and frequent partition – but it cannot be applied in smart grids: calculating the frequent group rekeying and storing additional $|A_{sm}|\cdot(\log_d T_t)$ (where $|A_{sm}|$: the number of attributes *sm* is given; *d*: the degree of a group key tree; T_t: the group size). KEKs are costly for each smart meter which retains limited memory storage and shows poor processing capability.

8. Conclusion

DR is a critical service in smart grids. However, its privacy leakage raises customers' concerns. DR data including utility messages and smart metering data can be easily mined to expose customers' privacy. We explore privacy violations and develop P2DR, a privacy-preserving protocol, to conceal customers' sensitive information with the use of an ABE scheme. At last, we present an efficient, periodic batch rekeying scheme to manage the ABE keys in case that smart meters join or leave. Experiments upon a simulated smart grid platform show that our P2DR incurs a significantly light performance overhead.

References

[1] M.H. Albadi and E.F. El-Saadany, *Demand response in electricity markets: An overview*, in *IEEE PES General Meeting*, 2007, pp. 1–5.
[2] P. Barreto, B. Lynn, and M. Scott, *Efficient implementations for pairing-based cryptography*, J. Cryptol. 17 (2004), pp. 321–334.
[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations among notions of security for public-key encryption schemes*, in *CRYPTO'98*, LNCS, vol. 1462, 1998, pp. 26–45.
[4] J. Bethencourt, A. Sahai, and B. Waters, *Ciphertext-policy attribute-based encryption*, in *IEEE S&P'07*, 2007, pp. 321–334.
[5] H.S. Cho, T. Yamazaki, and M. Hahn, *AERO: Extraction of user's activities from electric power consumption data*, IEEE Trans. Consum. Electron. 56 (2010), pp. 2011–2018.
[6] A.K. David and Y.Z. Li, *Consumer rationality assumptions in the real time pricing of electricity*, IEE Proc. Generat. Transm. Distrib. 139 (1992), pp. 315–322.
[7] C. Efthymiou and G. Kalogridis, *Smart grid privacy via anonymization of smart metering data*, in *IEEE SmartGridComm'10*, 2010, pp. 238–243.
[8] F.D. Garcia and B. Jacobs, *Privacy-friendly energy-metering via homomorphic encryption*, in *Security and Trust Management*, LNCS, vol. 6710, 2011, pp. 226–238.
[9] J. Hur and K. Kang, *Secure data retrieval for decentralized disruption-tolerant military networks*, IEEE/ACM Trans. Netw., in press.
[10] A. Lewko and B. Waters, *Decentralizing attribute-based encryption*, in *EUROCRYPT'11*, LNCS, vol. 6632, 2011, pp. 568–588.
[11] D. Li, Z. Aung, S. Sampalli, J.R. Williams, and A. Sanchez, *Privacy preservation for multicast communication in smart buildings of smart grids*, Smart Grid Renew. Energy, 4 (2013), pp. 313–324.
[12] D. Li, Z. Aung, J.R. Williams, and A. Sanchez, *P3: Privacy preservation for appliance control application*, in *Proceedings of the 3rd IEEE SmartGridComm*, 2012, pp. 294–299, Tainan, Nov.
[13] F. Li, B. Luo, and P. Liu, *Secure information aggregation for smart grids using homomorphic encryption*, in *SmartGridComm'10*, 2010, pp. 327–332.
[14] H. Li, R. Mao, L. Lai, and R.C. Qiu, *Compressed meter reading for delay-sensitive and secure load report in smart grid*, in *SmartGridComm'10*, 2010, pp. 114–119.

[15] D. Li and S. Sampalli, *A high performance contributory group key management scheme for resource-limited networks*, in *International Conference on Wireless Networks*, 2010, pp. 1–8.
[16] M.A. Lisovich, D.K. Mulligan, and S.B. Wicker, *Inferring personal information from demand-response systems*, IEEE Secur. Priv. 8 (2010), pp. 11–20.
[17] N. Liu, J. Chen, L. Zhu, J. Zhang, and Y. He, *A key management scheme for secure communications of advanced metering infrastructure in smart grid*, IEEE Trans. Ind. Elec. 60 (2013), pp. 4746–4756.
[18] B. Lynn, *The Stanford Pairing Based Crypto Library*. http://crypto.stanford.edu/pbc/
[19] S. McLaughlin, P. McDaniel, and W. Aiello, *Protecting consumer privacy from electric load monitoring*, in *ACM CCS'11*, 2011, pp. 87–98.
[20] Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.
[21] A.R. Metke and R.L. Ekl, *Security Technology for smart grid networks*, IEEE Trans. Smart Grid 1 (2010), pp. 99–107.
[22] NIST, *Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid*, NISTIR 7628, Aug 2010.
[23] NIST, *Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy*, Architecture, and High-Level Requirements, NISTIR 7628, Aug 2010.
[24] R. Ostrovsky, A. Sahai, and B. Waters, *Attribute-based encryption with non-monotonic access structures*, in *ACM CCS'07*, 2007, pp. 195–203.
[25] E.L. Quinn, *Privacy and the new energy infrastructure*, Social Science Research Network (SSRN), Feb (2009).
[26] S.R. Rajagopalan, L. Sankar, S. Mohajer, and H.V. Poor, *Smart meter privacy: A utility-privacy framework*, in *IEEE SmartGridComm'11*, 2011, pp. 190–195.
[27] A. Rial and G. Danezis, *Privacy-preserving smart metering*, in *ACM CCS Workshop WPES'11*, 2011, pp. 49–60.
[28] L. Sankar, S. Kar, R. Tandon, and H.V. Poor, *Competitive privacy in the smart grid: An information-theoretic approach*, in *IEEE SmartGridComm'11*, 2011, pp. 220–225.
[29] M. Tomosada and Y. Sinohara, *Virtual energy demand data: Estimating energy load and protecting consumers' privacy*, in *IEEE PES ISGT'11*, 2011, pp. 1–8.
[30] U.S. Department of Energy, Assessment of Demand Response and Advanced Metering. Federal Energy Regulatory Commission Report. Aug. 2006, http://www.FERC.gov
[31] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, *Time valid one-time signature for time-critical multicast data authentication*, in *IEEE INFOCOM'09*, 2009, pp. 1233–1241.
[32] S. Wicker and R. Thomas, *A privacy-aware architecture for demand response systems*, in *HICSS'11*, 2011, pp. 1–9.
[33] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel, *Minimizing private data disclosures in the smart grid*, in *ACM CCS'12*, 2012, pp. 415–427.
[34] J. Zhang and C.A. Gunter, *Application-aware secure multicast for power grid communication*, in *IEEE SmartGridComm'10*, 2010, pp. 339–344.

Appendix

Bilinear map

Bilinear map [2] works as the basis of our approach. $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ are a cyclic additive group and a cyclic multiplication group generated by P with the same order q, respectively. A mapping $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ satisfies the following properties:

- **Bilinear:** for all $u, v \in \mathbb{G}; a, b \in \mathbb{Z}_P$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$, where $=$ is an equation;
- **Computable**: there exists an efficient computable algorithm to compute $\hat{e}(u, v)$, $\forall u, v \in \mathbb{G}$;
- **Non-degenerate**: for the generator g of $\mathbb{G}$, p is the order of $\mathbb{G}$, we have $\hat{e}(g, g) \neq 1 \in \mathbb{G}_{\mathbb{T}}$;

Attribute-based encryption[4]

In this subsection, we list ABE's main primitives.

Access Tree – an access structure is represented by the tree T in which a leaf node is associated with a specific attribute and an intermediate node works as a 'AND' or 'OR' gate. We say that a set of attributes γ satisfies access tree if the root nodes' gate is true via recursively calculating roots' children nodes.

Algorithm 9. ABESet-Up(*PK, MSK*)

Input – *Null*
Output – *PK* : *Public key;*
MSK: Master secret key;
/* *This algorithm is generally executed by the key server* */
1: Randomly selects two credentials
$\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$
2: Calculates
$PK = \{ \mathbb{G}_0; g; h = g^{\beta}; f = g^{1/\beta}; e(g, g)^{\alpha} \}$;
$MSK = (\beta, g^{\alpha})$;

Algorithm 10. ABEKeyGen (*MSK*, $\mathbb{S}$, *SK*)

Input – *MSK*: *Master secret key*
$\mathbb{S}$: *A set of attributes*
Output – *SK* : Secret key
/* *This algorithm is generally executed by the key server* */
1: Generate a random $r \overset{R}{\leftarrow} \mathbb{Z}_p$
2: **for** $\{ \forall j \in \mathbb{S} \}$ /* *each attribute in* $\mathbb{S}$ */
3: Choose corresponding random $r_j \overset{R}{\leftarrow} \mathbb{Z}_p$
4: $SK = \{D = g^{(\alpha+r)/\beta};$
$\{\forall j \in \mathbb{S}:$
$D_j = g^r \times H(j)^{r_j}; \quad D_j^{'} = g^{r_j}\}$;
$\}$ /* *end of SK* */
The component of private keys is:
$D_i = g^{q(i)} T(i)^{r_i}$,
where $T(i) = g^{x^i} \prod_{j=1}^{n+1} t_j$
5: **end for**

Algorithm 11. ABEEncrypt (PK, M, T, CT)

Input – PK: *Public key;*
M *: Message ;*
$\mathcal{T}$: *Tree access structure;*
Output – CT: *Ciphertext;*
/ This algorithm is generally executed by the control server */*
1: **for** (each node x in the tree T)
2: selects a corresponding polynomial q_x ;
3: assigns its degree: $d_x = k_x + 1$;
where d_x *is the degree of polynomial* q_x *;*
k_x *is the threshold value of a node* x.
4: **end for** */* END for each node x in the tree T */*
5: **for** (nodes x on the tree T)
6: start at root and following the top-down manner:
7: **if** (node = root)
8: $q_R(0) = s$ *where* $s \in \mathbb{Z}_p$ *is a random.*
9: **else**
set $q_x(0) = q_{parent(x)}(index(x))$
where
parent(x) *returns node x's parent node*
index(x) *returns the ordering number of node*
x's sibling nodes. Ordering numbers are assigned
by x's parent node.
10: **end if**
11: randomly selects d_x other points for q_x to
complement the definition of the polynomial q_x.
12: **end for** */* END for nodes x on the tree T */*
13: Ciphertext is outputted as:
$CT = \{ \mathrm{T};\ \tilde{C} = Me(g,g)^{\alpha s};\ C = h^s;$
$\{\forall y \in Y :$
$C_y = g^{q_y(0)}; C_y^{'} = H(att(y)^{q_y(0)};\};$
$\}$
where function $att(x)$ returns attributes associated with the leaf node; Y: leaf nodes;
$H : \{0,1\}^{*} \rightarrow \mathbb{Z}_p$ is a collision-resistant hash function;

Algorithm 12. ABEDecrypt (PK, CT, SK, M)

Input – PK: *Public key;*
SK: *Secret Key*;
$\mathcal{T}$: *Tree access structure*;
CT: *Ciphertext*;
Output – M : *Message ;*
/* *This algorithm is generally executed by the smart meter* */
Function $DecryptNode(CT, SK, x)$
/* *The* $DecryptNode(CT, SK, x)$ *function below will be invoked recursively starting at root node R to verify if the access tree T can be satisfied by S:* */
if node x is a leaf node
  $i = att(x)$;
  if $i \notin S$,
    $DecryptNode(CT, SK, x) = \perp$
  else
    $DecryptNode(CT, SK, x) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} = \frac{e(g^r, g^{q_x(0)}) \cdot e(H(i)^{r_i}, g^{q_x()})}{e(g^{r_i}, H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)}$
  endif /* *END for* $i \notin S$ */
else /* *node x is not a leaf node* */
  for (all nodes $z \in$ node x's children nodes)
    $F_z = DecryptNode(CT, SK, z)$
    if ($F_z \neq \perp$)
      $F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x(0)}}$
      $= \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i,S'_x(0)}} = \prod_{z \in S_x} (e(g,g)^{r \cdot q_{parent(z)(index(x))}})^{\Delta_{i,S'_x(0)}} = \prod_{z \in S_x} \left(e(g,g)\right)^{r \cdot q_x(i) \cdot \Delta_{i,S'_x(0)}}$
      $= e(g,g)^{r \cdot q_x(i)}$
      where
        $i = index(z)$
        $S'_x = \{index(z) : z \in S_x\}$
    else /* $F_z = \perp$ */
    return $\perp$
    end if /* *END of* ($F_z \neq \perp$) */
  end for /* *END of all nodes x's children node* */
end if /* *END of node x is a leaf node* */
End Function /* *For* $DecryptNode(CT, SK, x)$ */
    R = Root node;
Call Function $DecryptNode(CT, SK, R)$
Decrypt ciphertext
$M = \frac{\check{C}}{e(C,D)/A} = \frac{\check{C}}{e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}}$