P3: Privacy Preservation Protocol for Automatic Appliance Control Application in Smart Grid

Depeng Li, Zeyar Aung, John Williams, and Abel Sanchez

***Abstract*—To address recently emerging concerns on privacy violations, this paper investigates possible sensitive information leakages and analyzes potential privacy threats in the automatic appliance control (AAC) application, which is one of the handiest applications in smart grids and one of the earliest examples in Internet of Things (IoT). Without an effective and consistent privacy preservation mechanism, the adversary can capture, model, and divulge customers' behavior, activities, and personal information at almost every level of society. Based on a set of existing cryptographic primitives, we propose an attribute-based encryption (ABE) key management variant and we also design and implement a fine-grained protocol named privacy preservation protocol (P3). We further present a practical automatic appliance control (AAC) system based on that protocol, and shows that it can fulfill the smart grid's requirements in privacy preservation. Experimental results demonstrate that our protocol merely incurs a substantially light overhead on the AAC application, yet is able to address and solve the formidable privacy challenges both customers and utility companies are facing.**

***Index Terms*—Attribute-based encryption, automatic appliance control (AAC), data privacy, privacy preservation.**

I. Introduction

THE Internet of Things (IoT) has become an emerging trend and a growing reality in our age. More ubiquitous and intelligence devices are embedded in our daily lives. As the first (and maybe largest) example of IoT, the smart grid is an intelligent electricity grid that possesses the capabilities to shave the power consumption peak, to optimize energy loss, to reduce customers' power bills, and to provide better power reliability [4]. The automatic appliance control (AAC) application, which is one of the handiest and most visible applications in IoT deployment in the smart grid, is widely utilized by end customers and the utility companies. However, the digitization movement to replace "dumb" devices (e.g., meters) with smart devices (e.g., smart meters) creates an intrinsic link between electricity customers and those smart devices. The smart grid generates and archives large volumes of high-resolution smart grid data such as power consumptions, control commands, events, and alarms. These data could be potentially used (or misused) beyond their original purposes for which they are collected. For example, one could illegally use them to reveal customers' daily activities and individual behavior models [16], [30]. Some pioneering studies, e.g., [28], explored means to profile human activities by converting the power consumption data into a timeline of appliance uses.

On the other hand, there have been only very few discussions regarding privacy leakage from AAC applications. Their privacy risks exist just as other human-centric activities involving personal data. Direct access to such data can easily infer users' activity patterns. For example, remote AAC commands that shut down air conditioners (A/C) in a particular dwelling when the temperature is high (e.g., $>104^\circ\mathrm{F}/40^\circ\mathrm{C}$) may be highly indicative of the current absence of their residents. Furthermore, some potential exploitations of AAC data maybe still unknown nowadays, however critical they may be decades later. Thus, privacy protection schemes to hide AAC commands and to eliminate personal information are highly desirable.

This paper explores the privacy preservation for AAC applications in smart grids by investigating the benefits from cryptographic methods. We construct a privacy protection mechanism based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [6] and Rivest, Shamir, and Adleman (RSA) public key (PK) encryption algorithms [32] to avert privacy disclosure for AAC applications.

However, utilizing the original ABE system in AAC applications poses some formidable challenges, namely, the needs for the following.

1) *Key management for ABE scheme:* In the smart grid, it is possible that some smart devices, e.g., smart meters are expired or out-of-service. Meanwhile, new smart meters will be installed. Without the satisfaction of forward secrecy and backward secrecy, ABE key schemes may lead to the potential security vulnerability.
2) *Efficiency of ABE key management:* Smart devices, e.g., smart meters have constraint resources, e.g., limited computational capacity and smart grid communications demonstrate confined bandwidth. Thus, efficiency is highly demanded.
3) *Well-designed attribute management system:* Each smart meter should be assigned a set of predefined attributes in such a way that smart meters are uniquely identified and effectively managed.

Our Contributions: Our research carefully studies privacy leakages and provides a privacy preserving protocol for AAC in smart grids, which is one of the first examples for IoT

Manuscript received August 04, 2014; accepted September 10, 2014. Date of publication September 17, 2014; date of current version October 21, 2014. An earlier version of this paper was presented at the 3rd IEEE International Conference on Smart Grid Communications, 2012.

D. Li is with the Department of Information and Computer Sciences, University of Hawaii at Manoa, Honolulu, HI 96822 USA (e-mail: depengli@hawaii.edu).

Z. Aung is with the Department of Electrical Engineering and Computer Science, Masdar Institute of Science and Technology, Abu Dhabi 54224, United Arab Emirates (e-mail: zaung@masdar.ac.ae).

J. Williams and A. Sanchez are with the Engineering Systems Division, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: jrw@mit.edu; doval@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2358578

deployment. First, we discuss the possible privacy leakages related to AAC such as the list of automatically controlled electrical appliances that residents own, the status of those appliances, and residents' absence/presence.

Second, we develop and implement a fine-grained protocol, namely, *privacy preservation protocol (P3)* through the usage of cryptographic primitives as well as some adapted and augmented countermeasures. In P3, the control server multicasts smart meters, i.e., the control messages that will be granularly encrypted by ABE system so that the messages cannot be accessed by any nondesignated smart meters. We use the *address system*, a descriptive and effective means, to identify the attributes of smart meters in customers' premises.

Third, we propose a new key management for the ABE system based on our previous research on periodic rekeying scheme [22]. It fixes the problem introduced by previous ABE key revocation schemes which cannot accommodate newly installed smart meters in an efficient and secure way. We further utilize the efficient periodic batch rekeying strategy which is desirable for resource-limited smart meters.

Finally, our prototype was executed on commodity control servers and emulated smart meters. The experimental results demonstrated that our solution merely incurs a low delay (<440 ms) which is acceptable to AAC application in the smart grid. Computational cost by P3 is very light weight and it exhibits efficient performance even on the emulated smart meters, which are configured with low-end central processing unit (CPU) and limited memory, in our experiment.

II. Background and Problem Description

A. Symbol Table

A symbol table is reflected here as Table I.

B. Examples of AAC

Automatically controlling smart appliances such as A/C, pool pumpers, and dishwashers by the smart grid, by the short message service (SMS), or even locally by a personal computer (PC) at home is not only possible but also becomes a common trend in smart homes nowadays [2], [3]. AAC has been utilized in three subareas: 1) demand–response program [37] provided by smart grid systems; 2) customer remote control service [3] supplied by utility companies through secure channels via Internet; and 3) home appliance automatic control system [2]. In all of them, control commands or status messages are transferred through public communication services such as Internet and telecommunications, which may probably confront the potential threat of the eavesdropping attack if no specific security scheme is deployed.

Two examples describing the AAC in smart grids from different perspectives are as follows.

1) *Direct load control program:* Demand response (DR) programs aim to balance the supply and the load in real time. The direct load control program is a classical DR which enables utility companies to remotely shut down residence's appliances in a short notice with customers' prior consent, while the system in jeopardy is sensed [37].

TABLE I
SYMBOL TABLE

$\{a\}b$	a is encrypted by b
$\overset{R}{\leftarrow}$	randomly generate
$\rightarrow$	forward
$attr_i$	attributes of access policy in ABE
a, b	$a, b \in \mathbb{Z}$
$\hat{A}$	attribute set of a smart meter
A	DecryptNode(CT,SK,R)
C_j	control command with one or more than one receipt
$\tilde{C}, C_y, C_y', C$	part of ciphertext CT in ABE
$Comp()$	server processing cost for key tree in rekeying
CS	a control server in utilities
CT	ciphertext
d	degree of key tree used in H&K and our ABE variant
d_x	degree of polynomial q_x in ABE
D, D_j, D_j'	part of secret key SK in ABE
$\text{Đ}, \text{Đ}_j, \text{Đ}_j'$	part of secret key SK' in our ABE
D_{RSA}	computational cost of RSA decryption
$\hat{e}, e$	bilinear mapping
E_{RSA}	computational cost of RSA encryption
f	part of public key (PK) in ABE
g	generator of $\mathbb{G}$
G_{key}	group key used in our proposal
h	part of public key (PK) in ABE
H	a collision-resistant hash function
i	index
j	an attribute in ABE
J	the number of joining smart meters in group rekeying interval
k_x	threshold value of a node x in access tree $\mathcal{T}$ in ABE
k_{λ_y}	group key corresponding to attribute λ_y in H&K[12]
L	the number of leaving smart meters
m	number of multicast commands in a list M
M	multicast message encapsulating control commands
MK	master secret key of ABE
n	number of smart meters in a group
N	number of legal smart meters for group rekeying
PK	public key
q	order of $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$
q_x	polynomial
r	a random
r_j	a random corresponding to an attribute j in ABE
r_{new}, r_j^{new}	random in group rekeying
R	a set of join/leave requests in a periodic rekey interval
RT	the root node in access tree $\mathcal{T}$ in ABE
s	a random
S	a set of attributes in ABE
S_x	an arbitrary set of child nodes in access tree $\mathcal{T}$ in ABE
SK, SK'	secret key in ABE
sm_i	smart meter i
t	height of key tree
u, v	$u, v \in \mathbb{G}$
W	a matrix includes a number of attributes
x	a node in access tree $\mathcal{T}$ in ABE
x_{ij}	an attribute for smart meter
y	$\forall y \in Y$
z	a node in access tree $\mathcal{T}$ in ABE
α	$\alpha \in \mathbb{Z}_p$
β	$\beta \in \mathbb{Z}_p$
γ	a set of attributes
$\Delta_{i,S}$	lagrange coefficient for $i \in \mathbb{Z}_p$and S, a set of elements
$\mathbb{G}$	cyclic additive group
$\mathbb{G}_{\mathbb{T}}$	a cyclic multiplication group
$\mathcal{T}$	access tree structure

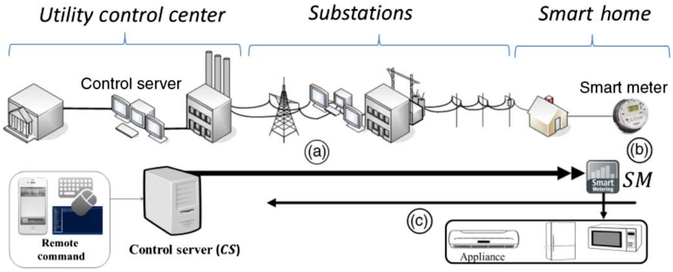

Fig. 1. Model of AAC. (a) A control server multicasts a command list: $M = \{\{C_1\} \dots \{C_m\}\}$, where $|M| = m$. (b) Smart meter executes the command C_j, where $1 \leq j \leq m$ to control its appliance if one of C_j's recipients is smart meter sm_i. Note that C_j may apply to a number of smart meters. (c) After executions of C_j, smart meter sm_i unicasts back results to the control server.

The remotely controlled devices can be smart appliances such as A/C and water heaters.

2) *Remote AAC:* Currently, some utility companies provide customers with capabilities to remotely control appliances through the smart grid. An explicit control is enabled by interactions between the smart grid and the user through smart phones, websites, or even e-mails. For example, a person can remotely turn ON A/C in his home before going home from work. Then, upon arriving at home, he can relish the cool air. Another example is that a customer may forget to turn OFF the appliances such as pool pumpers and A/C before he leaves home. But, he can later turn them OFF remotely from outside in order not to waste electricity.

C. AAC Model

Fig. 1 demonstrates the AAC application in smart grids which comprises the control server (CS) (deployed in the command and control center in a utility company), a number of smart meters, sm (deployed in residence and can control the appliances), and the communication channels in between. The smart meters can control the smart appliances based on the remote control commands from customers or utilities.

In the smart grid, multicast communication is extensively deployed because of its scalability, efficiency, and functionality across network segments [25], [40]. AAC applications can also utilize multicasting for the sake of efficiency.

We assume that a residence has a smart meter sm_i installed at it. A residential address is used in our system to identify a smart meter sm_i which is represented by an attribute set such as $\widehat{A} = \{\mathrm{attr1} =$ "house number"; $\mathrm{attr2} =$ "street name"; $\mathrm{attr3} =$ "ZIP value"; $\mathrm{attr4} =$ "city name"$\}$.

As depicted in Fig. 1, the control server multicasts smart meters a command list M which encapsulates a set of commands, e.g., $\{\dots, C_j, \dots\}$. After receiving M, a smart meter sm_i can be aware that C_j is designated for itself if the attributes of sm_i satisfy C_j's access policy. Then, after executing C_j, sm_i sends the result to the control server.

D. Problem Description of Privacy Threats

In this paper, we discover that the privacy threat [28] can occur when an adversary associates an AAC command C_i with personal information, e.g., customers' private information, activity models, and preferences.

1) *Privacy of Residence Occupancy:* An AAC command C_j can let an adversary infer whether a resident is present or absent. (This is also referred to as *absence privacy*.)

a) *Example I:* Alice sends a remote control command to "address $\widehat{A}$" aiming to shut down its A/C, when the local temperature outdoor is high (e.g., $>104°\mathrm{F}/40°\mathrm{C}$). Eve can probably infer that residence with "address $\widehat{A}$" may possibly be empty and then he can take the risk to break in.

2) *Privacy of Appliance Ownership:* The history of AAC commands $\{\dots, C_j, \dots\}$ let adversaries compile a list of household appliances and surmise the lacking one.

a) *Example II:* Alice had sent home the remote AAC commands associated with A/C, heater, and washing machine but not dishwasher. Eve extrapolates that it is highly possible for Alice to not own a dishwasher yet. This information can be commercially valuable; advertisements of dishwashers can be targeted to Alice.

3) *Privacy for Personal Activity Model:* The AAC commands $\{\dots, C_j, \dots\}$ can let the adversary generalize the resident's activity model.

a) *Example III:* Alice always remotely turns ON his A/C half an hour earlier before arriving at home. Eve finds that these control commands are sent out at 5:30 P.M. from every Tuesday to Friday but 6:30 P.M. every Monday. Eve can guess Alice's work and life schedules based on it.

E. Cryptography Primitives

1) *Bilinear Map:* Bilinear map [5], [8] 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 $\hat{e}\colon \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ satisfies the following properties.

- *Bilinear:* for all $u, v \in \mathbb{G}; a, b \in \mathbb{Z}$, 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 $\hat{e}(u, v), \forall u, v \in \mathbb{G}$.
- *Nondegenerate:* for the generator g of $\mathbb{G}$, q is the order of $\mathbb{G}$, we have $\hat{e}(g, g) \neq 1 \in \mathbb{G}_{\mathbb{T}}$.

2) *Attribute-Based Encryption (ABE) [6]:*

a) *Access tree:* An access structure is represented by the tree in which a leaf node is associated with a specific attribute and an intermediate node works as an "AND" or "OR" gate. We say that a set of attributes γ satisfies access tree $\mathcal{T}$ if the root nodes' gate is true via recursively calculating roots' children nodes.

Setup() $\rightarrow$ (PK, MK);
/* *public key* PK; *master secret key* MK; */

- Randomly selects two credentials $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p$;
- Calculates

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

Key Generation $(\mathrm{MK}, S) \rightarrow \mathrm{SK}$

/* MK master key; a set of attributes S; Secret key SK */

- Generate a random $r \overset{R}{\leftarrow} \mathbb{Z}_p$.

For each attribute $j \in S$,

- Choose corresponding random $r_j \overset{R}{\leftarrow} \mathbb{Z}_p$
- Calculate

$$\begin{aligned}\mathrm{SK} = \{&D = g^{\frac{\alpha+r}{\beta}};\\ &\{\forall j \in S:\\ &\quad D_j = g^r \times H(j)^{r_j};\ D'_j = g^{r_j}\};\\ &\}\end{aligned}$$

- For all $i \in \mathcal{T}$ the private keys components are

$$\begin{aligned}D_i &= g^{q(i)} T(i)^{r_i},\\ d_i &= g^{r_i}\end{aligned}$$

where $T(i) = g^{x^i} \prod_{j=1}^{n+1} t_j{}^{\Delta_{j,N}(i)}$

Encrypt $(\mathrm{PK}, M, T) \rightarrow \mathrm{CT}$

/* *public key* PK; *message* M; *tree access structure* $\mathcal{T}$ *ciphertext* CT */

- For each node x in the tree $\mathcal{T}$, select a corresponding polynomial q_x then assign its degree: $d_x = k_x + 1$ where d_x is the degree of polynomial q_x and k_x is the threshold value of a node x.
- Beginning at the root node RT, first assigns $q_R(0) = s$ where $s \in \mathbb{Z}_p$ is a random. Second, randomly selects d_R other points for q_R to complement the definition of the polynomial q_R.
- Process the rest nodes x on the tree T by following the top-down manner: sets $q_x(0) = q_{\mathrm{parent}(x)}(\mathrm{index}(x))$ where function $\mathrm{parent}(x)$ returns node x's parent node and function $\mathrm{index}(x)$ returns the ordering number of node x's sibling nodes. Ordering numbers are assigned by x's parent node. Then, randomly selects d_x other points for q_x to complement the definition of the polynomial q_x.
- Ciphertext is output as

$$\begin{aligned}\mathrm{CT} = \{&\mathcal{T}; \tilde{C} = \mathrm{Me}(g,g)^{\alpha s};\ C = h^s;\\ &\{\forall y \in Y:\\ &\quad C_y = g^{q_y(0)};\ C'_y = H(\boldsymbol{att}\,(y)^{q_y(0)});\};\\ &\}\end{aligned}$$

where function ***att*** $(\boldsymbol{x})$ returns attributes
associated with the leaf node;
$H: \{0,1\}^* \rightarrow \mathbb{Z}_p$ is a collision-resistant
hash function;

Decrypt $(\mathrm{PK}, \mathrm{CT}, \mathrm{SK}) \rightarrow M$

/* *Public Key* PK*:* *Ciphertext* CT; *Private key* SK; */

The $DecryptNode(CT, SK, x)$ function below will be invoked recursively starting at root node RT to verify if the access tree T can be satisfied by S.

- If the node x is a leaf node, set $i = \boldsymbol{att}(x)$:

```
if $i \notin S$
    $DecryptNode(CT, SK, x) = \perp$
if $i \in S$
```

$$\begin{aligned}&DecryptNode(CT, SK, x)\\ &\quad= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})}\\ &\quad= \frac{e(g^r, g^{q_x(0)}) \cdot e(H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})}\\ &\quad= e(g,g)^{r q_x(0)}. \qquad (1)\end{aligned}$$

- If the node x is not a leaf node.

For all nodes z which are node x's children nodes, call function $F_z = DecryptNode(CT, SK, z)$. Assign S_x with an arbitrary k_x-sized set of child nodes in such a way that $F_z \neq \perp$. If we cannot find such set, it means that the node cannot be satisfied, and the function returns $\perp$.

Otherwise, calculate

$$\begin{aligned}F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,s'_x}(0)} = \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_z(0)}\right)^{\Delta_{i,s'_x}(0)}\\ &= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_{\mathrm{parent}(z)}(\mathrm{index}(x))}\right)^{\Delta_{i,s'_x}(0)}\\ &= \prod_{z \in S_x} (e(g,g))^{r \cdot q_x(i) \cdot \Delta_{i,s'_x}(0)} = e(g,g)^{r \cdot q_x(i)} \qquad (2)\end{aligned}$$

where $i = \mathrm{index}(z)$, $S'_x = \{\mathrm{index}(z) : z \in S_x\}$, and $\Delta_{i,s'_x}(0) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$.

- Decrypt ciphertext

$$\frac{\check{\mathrm{C}}}{e(C,D)/A} = \frac{Me(g,g)^{\alpha s}}{e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}} = M \qquad (3)$$

where $A = DecryptNode(CT, SK, R)$.

III. System Overview

We first investigate the adversary model, discuss the security assumptions, and define the scope of this paper. Then, we introduce our system goal as well as our system overview.

A. Adversary Model, Security Assumptions, and Scope

1) Adversary Model: Like other researches in areas of privacy preservations [11], [12], [17], [21], [36], we follow the semihonest adversary model in which smart devices (e.g., smart meters) obey AAC schemes. Meanwhile, they are also curious about the messages they learn (or share) and have the intension to combine these information if possible. Therefore, any participating smart devices should relay packets and also intend to uncover others' privacy by studying sensitive messages they received.

2) Scope and Security Assumptions: Our protocol mainly focuses on the confidentiality service for communications between smart meters and utilities aiming to protect privacy. Other security properties such as integrity and authentication services (e.g., our previous research on authentication for the smart grid [20]) are also important but beyond this paper's scope. The two ends of smart grid communication channels can be vulnerable. There are some attacks against smart meters [31] and the utility companies can be compromised. However, due to the limited space of this paper, the trustworthiness of utilities, the physical security of smart meters, and the privacy protection between smart meters and appliances (e.g., our research on privacy within customer premises [19]) are out of the scope of this research. Therefore, we assume that smart devices such as smart meters are tamper-resistant and device attestations are deployed to validate them in this paper. Furthermore, we also assume that the utilities deploy the PK Infrastructure (PKI) and trusted

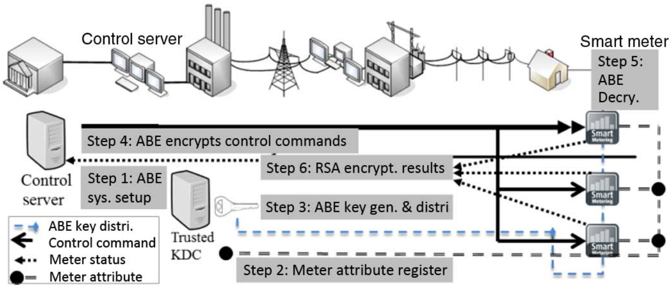

Fig. 2. Our system model.

Key Distribution Center (KDC) [32]. To avoid the vulnerability that the confidentiality critically depends on the security of a single-trusted KDC, CP-ABE can be easily replaced with multi-authority CP-ABE [18] which will be integrated into our future work. In this paper, we assume that the KDC is trustworthy. Likewise, we assume that each smart device holds its private key and publish PK.

B. System Goal and Overview

The goal of our proposed solution is to realize an efficient security mechanism to prevent privacy exposures while satisfying scalability and time-critical requirements of the AAC application in smart grid.

As depicted in Fig. 2, there are three participants in our system: 1) smart meters installed in the customers' residences; 2) control servers; and 3) trusted KDC deployed in utility control centers. To protect the multicast communication which sends crucial AAC commands from the control server to multiple smart meters in the system, we adopt an ABE encryption system [6] and propose our ABE variant which satisfies the backward and forward secrecies in P3. To screen the unicast which feedbacks results from a smart meter to the control server, we deploy the RSA public key encryption system for the sake of computational efficiency. The KDC's responsibility is to issue ABE keys and RSA private key/PK pairs to control servers and smart meters.

IV. Privacy Preservation Protocol

Our protocol, P3, conceals sensitive data transmitted in AAC applications via integrating the cryptographic primitives, e.g., the ABE [6], and in conformance with smart grid's regulations.

P3 is illuminated in Fig. 3. The control server multicasts smart meters the AAC command list M in which each command entry C_j is encrypted by the ABE encryption algorithm with the PK and the access policy $\mathtt{attr}_j$ [as depicted at the bottom of Fig. 3(b), the access policy is hidden to protect contextual privacy]. Each smart meter sm decrypts the ciphertext by using its own secret key SK if its attributes attr satisfies access policy $\mathtt{attr}_j$. After executing the command entry C_j, sm encrypts the operational result with the control server's RSA PK and sends it back. The control server decrypts it with its RSA secret key.

A. New Contributions of P3

The smart grid exhibits its unique features. 1) The smart meters are relatively static over time. When utility companies plan to install the smart grid in an area, there will be rare *leave* (e.g., expired/replaced) events but a number of *join* (e.g., new installation) events in a specific rekeying interval. 2) Smart meter membership activities (e.g., *join*/*leave*) are strictly regulated by utilities. 3) Smart meters are resource-limited in terms of memory storages and processing capabilities and meanwhile, the smart grid communication bandwidth is restricted [20], [23], [25]. 4) Both backward and forward secrecies are mandatory [39]. P3 satisfies these requirements as follows.

1) Periodic batch rekeying scheme for ABE: Our periodic rekeying scheme processes the *join*/*leave* requests in batches. It achieves a significant performance improvement. Moreover, our scheme satisfies the backward secrecy and the forward secrecy: it accommodates the valid smart meter events (e.g., new-join and leave) in such a way that newly joined smart meters cannot decrypt previous messages via using the newly issued secret keys SK and the leaving smart meters cannot decrypt the subsequent AAC commands.
2) Smart meters send its operational result/status back to the control server via the RSA encryption, after the execution of commands they received. The reason is because RSA is more computationally efficient than ABE encryption and it is validated by our experimental results in Section VI.
3) The access policies associated with AAC commands are efficiently hidden by the one way hash function.

B. Discussion of the Selection of Cryptographic Schemes

1) Periodic Batch Rekeying: In individual rekeying, each membership change request, e.g., joining/leaving will be processed immediately. In contrast, periodic batch rekeying collects those requests and will not process them until the end/start of an interval. Although individual rekeying is ideal, there are two problems [26].

a) Inefficiency: Individual rekeying has to immediately response to each *join*/*leave* request launched by the smart meter. The corresponding operations are processed to accomplish rekeying: to generate a new group key and to deliver the encrypted key encryption key (KEK) to each member. It is easy to understand its efficiency: In case of J *join* and L *leave* occurred in a rekeying interval, there will be $(J + L)$ rekeying operations; in contrast, periodic rekeying only requires *one*. Considering the low-end configurations of smart meters and limited communication bandwidth in smart grids, the expensive cryptographic operations and heavy communication traffic could be a big challenge. It also lacks productivity: when *two* membership change requests happen very close to each other with a quite small time interval, the first set of rekeying could not be used and then it is replaced by the second set. It is waste of resources ranging from CPU to communication bandwidth.

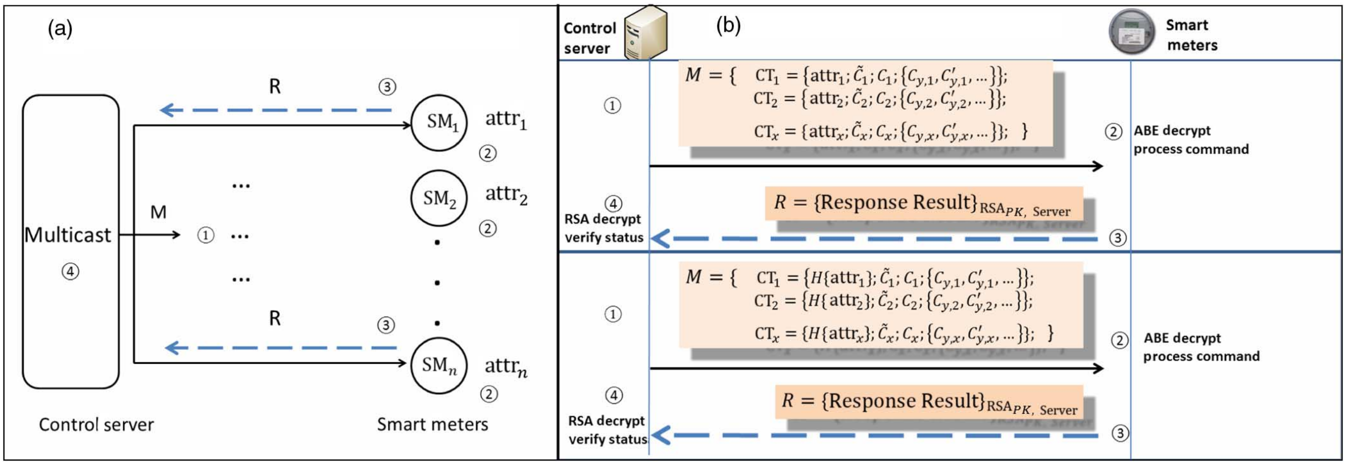
(a)
R
③
SM_1
attr_1
②
...
M
Multicast
④
①
...
SM_2
attr_2
②
...
R
③
SM_n
attr_n
②
Control server
Smart meters
(b)
Control server
Smart meters
①
$M = \{ \text{CT}_1 = \{\text{attr}_1; \tilde{C}_1; C_1; \{C_{y,1}, C'_{y,1}, ...\}\};$
$\text{CT}_2 = \{\text{attr}_2; \tilde{C}_2; C_2; \{C_{y,2}, C'_{y,2}, ...\}\};$
$\text{CT}_x = \{\text{attr}_x; \tilde{C}_x; C_x; \{C_{y,x}, C'_{y,x}, ...\}\}; \}$
② ABE decrypt process command
④ RSA decrypt verify status
$R = \{\text{Response Result}\}_{\text{RSA}_{PK,\ \text{Server}}}$
③
①
$M = \{ \text{CT}_1 = \{H\{\text{attr}_1\}; \tilde{C}_1; C_1; \{C_{y,1}, C'_{y,1}, ...\}\};$
$\text{CT}_2 = \{H\{\text{attr}_2\}; \tilde{C}_2; C_2; \{C_{y,2}, C'_{y,2}, ...\}\};$
$\text{CT}_x = \{H\{\text{attr}_x\}; \tilde{C}_x; C_x; \{C_{y,x}, C'_{y,x}, ...\}\}; \}$
② ABE decrypt process command
④ RSA decrypt verify status
$R = \{\text{Response Result}\}_{\text{RSA}_{PK,\ \text{Server}}}$
③

Fig. 3. Overview of the P3. (a) Overview of privacy preservation protocol. (b) Plain attr versus hashed attr.

b) Out-of-Sync: In not only art but also practice, the out-of-sync problem challenges the smart grid systems regarding keys and data. Due to delay of messages in a smart grid communication, a smart meter sm may receive a control message C_j encrypted by an old key $G_{\text{key,old}}$ or C_j is encrypted by current group key but sm has not received the current group key yet. Therefore, in the individual rekeying, each smart meter has to buffer a number of group keys as well as a lot of encrypted control commands. All this will cost huge amount of memory which may not be a good choice for resource-limited smart meters.

In contrast, the periodic batch rekeying can not only be efficient but also alleviate the out-of-sync problem since if each rekeying interval is big enough, those issues can be mitigated. However, the periodic batch rekeying introduces the *vulnerability window* which is defined as a period of time starting at the first join/leave request and ending at the end of the rekeying interval. Since the membership change request will not be processed immediately, the leaving/expelled smart meter can stay longer and the new installed smart meter has to join later. In a vulnerability window, the departure smart meter can still decrypt the encrypted control messages even though the time is somewhat short. According to [26], the vulnerability window is not only applied to periodic rekeying but also to the individual rekeying: for the latter, the vulnerability windows start when the *leaving* requests are sent and end at the time when all smart meters receive the new key. It includes the following procedures: 1) the leaving request is forwarded to the server; 2) the server processes the request; 3) the server processes the rekeying operation via generating new keys; 4) new keys are forwarded to all smart meters; and 5) smart meters decrypt the cipher to extract the new key. Of course, the vulnerability of individual rekeying is much shorter than that of periodic rekeying.

2) ABE and RSA: To reduce the communication overhead, we invoke ABE encryption while multicasting control commands which may apply to a lot of smart meters. In contrast, if we utilize *1-to-1* encryption scheme, e.g., RSA encryption, n times cryptographic operations are required but ABE only demands *one*. When a smart meter sends back its operational results to the server via the unicast, RSA encryption algorithm rather than ABE is utilized since RSA encryption is more efficient: the former lasts 8 ms and the latter around 200 ms (with one attribute) based on our experimental result captured upon our simulated smart meters. A similar ratio also gains according to experiments conducted upon a laptop [38].

C. Proposed ABE Key Management in P3

The original ABE system [6] or its variants [18], if deployed on the smart grid, will revoke expired ABE keys for smart meters using expiry dates/times. But they will not be able to efficiently handle newly joined smart meters with *backward privacy* service. The immediate key revocation scheme is proposed through the use of negative clauses [34]. However, its performance will not be efficient for smart meters. The attribute revocation scheme [12] utilizes the group key scheme (e.g., [22]) to generate attribute group keys, e.g., k_{λ_y} through which to encrypt and to deliver the updated ABE secret keys. However, as a sophisticated ABE key management scheme, the scheme in [12] is designed for the battlefield where the devices carried by soldiers may expect frequent attribute changes and the security is highly demanded. If we utilize it directly in P3, each smart meter may suffer severe performance issues and out-of-sync problems due to smart meters' low-end configurations.

Now, we propose a periodic batch ABE rekeying scheme which provides both *backward privacy* and *forward privacy*: All *new-join* and *leave* requests from smart meters are processed in a batch at the end of each rekeying interval rather than being processed individually. Our scheme is described below but our ABE setup is skipped since it stays the same as that in the original ABE scheme. Note that our ABE scheme follows the semihonest model introduced early: the server and the smart meters will follow our ABE scheme to encrypt/decrypt as well as forward messages but they also intend to combine each other's information to uncover the ciphertext which they cannot decrypt by themselves.

1) Collect Join/leave Requests: The trusted KDC accumulates *join/leave* requests from smart meters sm in the interval of a rekey period

$$\text{sm} \rightarrow \text{KDC}\colon R = \{\text{Request} \| S_i\}$$

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

2) Mark Affected Smart Meters: For smart meters which join/leave in this period, the trusted KDC enumerates all of their attributes and put them all in W as they are the affected attributes that need to be renewed

$$W = \begin{pmatrix} S_1 \\ \ldots \\ S_n \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1 \cdot m_1} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{n \cdot m_n} \end{pmatrix} \tag{4}$$

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

Then, we mark each smart meter that has one of its attributes belonging to W. That is, for a smart meter sm and its attribute set S_i, we verify whether there exists $x_{ij} \in W$ where $\forall x_{ij} \in S_i$. If so, we mark that smart meter sm.

3) Proposed ABE Batch Key Update: At the end of the rekeying period, the trusted KDC will 1) generate a group key G_{key} and 2) update the smart meter sm's secret key SK if sm is marked.

In our solution, there are two methods to generate the G_{key}: (a) to randomly compute the value and (b) to calculate a group key via using keygem scheme [22], [39]. If the (a) method is used, RSA encryption channels should be deployed to deliver the new ABE secret keys to each smart meters point to point. If the (b) method is used, the group key will be generated via constructing key tree and taking advantage of the auxiliary keys as shown in [39]. In that case, the multicast integrating with the key tree will be used to forward the new ABE secret key. Note that, in the (b) method, G_{key} and auxiliary keys in the key tree will be shared by current smart meters including the new installed ones. Here, we will not describe the detailed algorithms/group key agreement due to space limits. Refer to [39] for details.

The (a) method shows lightweight computational cost since it only generates one random value. But its communication overhead is large. In contrast, the (b) method demands complicated computational operations to create and to maintain the key tree, but its communication overhead is light due to the utilization of multicast. Detailed performance assessment is provided in Section VI to compare (a) with (b).

Then, we introduce how to calculate the ABE secret key SK for smart meters as follows.

For impacted smart meters:

- Generate a random $r_{\text{new}} \overset{R}{\leftarrow} \mathbb{Z}_p$.
- For each attribute $x_{ij} \in S_i$

  if $x_{ij} \in W$, create a random $r_j^{\text{new}} \overset{R}{\leftarrow} \mathbb{Z}_p$
- Calculate: SK'

$$= \left\{ \text{Đ} = g^{\frac{\alpha + r_{\text{new}}}{\beta}} G_{\text{key}}; \left\{ \forall x_{ij} \in S_i : (\text{Đ}_j, \text{Đ}'_j) \right. \right.$$

$$\left. \left. = \begin{cases} \text{Đ}_j = g^{G_{\text{key}} r_{\text{new}}} \times H(j)^{r_j^{\text{new}}}; \text{Đ}'_j = g^{r_j^{\text{new}}} \; x_{ij} \in \text{W} \\ \text{Đ}_j = g^{G_{\text{key}} r_{\text{new}}} \times H(j)^{r_j}; \text{Đ}'_j = g^{r_j} \; x_{ij} \notin \text{W} \end{cases} \right\} \right\}$$

For nonimpacted smart meters:

- Calculate: $\text{SK}' = \left\{ \text{Đ} = g^{\frac{\alpha + r}{\beta}} G_{\text{key}}, \right.$

$$\left. \left\{ \forall j \in S : \text{Đ}_j = g^{G_{\text{key}} r_{\text{new}}} \times H(j)^{r_j}; \text{Đ}'_j = g^{r_j}, \right\} \right\}$$

4) Proposed ABE Key Delivery:

(a) If G_{key} is a randomly generated value:

The trusted KDC encrypts the updated ABE secret key SK with the corresponding smart meter sm_i's RSA PK RSA_{sm} and delivers the ciphertext to sm_i

$$\text{KDC} \rightarrow \text{sm}_i : C = \{\text{SK}'\}\text{RSA}_{\text{sm}}.$$

So, the secret key of each marked smart meter is updated.

(b) If G_{key} is a group key generated via using key tree:

Refer to [22] and [39] for details about how the auxiliary keys and multicast are integrated together to distribute the secret information in format of ciphertext.

5) Proposed ABE Encryption:

$$\begin{aligned} \text{CT} = \{ & \mathcal{T}; \tilde{C} = Me(g,g)^{G_{\text{key}} \alpha s}; C = h^s; \\ & \{\forall y \in Y : \\ & \quad C_y = g^{q_y(0)}; C'_y = H(\text{att}(y)^{q_y(0)}); \}; \\ & \} \end{aligned}$$

6) Proposed ABE Decryption

Decrypt $(\text{PK}, \text{CT}, \text{SK}) \rightarrow M$

/* *Public Key* PK*: Ciphertext* CT; *Private key* SK; */

The $DecryptNode(CT, SK, x)$ function below will be invoked recursively starting at root node RT to verify if the access tree T can be satisfied by S.

- If the node x is a leaf node, set $i =$ ***att*** (x):

  if $i \notin S$

$$DecryptNode(CT, SK, x) = \perp$$

  if $i \in S$

$$\begin{aligned} & DecryptNode(CT, SK, x) \\ & = \frac{e(\text{Đ}_i, C_x)}{e(\text{Đ}'_i, C'_x)} = \frac{e\left(g^{G_{\text{key}} r_{\text{new}}} \cdot H(i)^{r_j^{\text{new}}}, g^{q_x(0)}\right)}{e\left(g^{r_j^{\text{new}}}, H(i)^{q_x(0)}\right)} \\ & = \frac{e\left(g^{G_{\text{key}} r_{\text{new}}}, g^{q_x(0)}\right) \cdot e(H(i)^{r_j^{\text{new}}}, g^{q_x(0)})}{e\left(g^{r_j^{\text{new}}}, H(i)^{q_x(0)}\right)} \\ & = e(g,g)^{G_{\text{key}} r_{\text{new}} q_x(0)}. \end{aligned} \tag{5}$$

- If the node x is not a leaf node:

For all nodes z which are node x's children nodes, call function $F_z = DecryptNode(CT, SK, z)$. Assign S_x with an arbitrary k_x-sized set of child nodes in such a way that $F_z \neq \perp$. If we cannot find such set, it means that the node cannot be satisfied, and the function returns $\perp$.

Otherwise, calculate

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i, s'_x}(0)} = \prod_{z \in S_x} \left(e(g,g)^{r_{\text{new}} \cdot q_z(0)}\right)^{\Delta_{i, s'_x}(0)} \\ &= \prod_{z \in S_x} \left(e(g,g)^{r_{\text{new}} \cdot q_{\text{parent}}(z)(\text{index}(x))}\right)^{\Delta_{i, s'_x}(0)} \\ &= \prod_{z \in S_x} (e(g,g))^{r_{\text{new}} \cdot q_x(i) \cdot \Delta_{i, s'_x}(0)} = e(g,g)^{r_{\text{new}} \cdot q_x(i)} \end{aligned} \tag{6}$$

where $i = \text{index}(z)$ and $S'_x = \{\text{index}(z)\colon z \in S_x\}$.

- Decrypt ciphertext

$$\frac{\hat{C}}{e(C,\text{Đ})/A} = \frac{Me(g,g)^{\alpha s G_{key}}}{e\left(\left(g^{\beta}\right)^{s}, g^{(\alpha+r_{\text{new}})G_{\text{key}}/\beta}\right)/e(g,g)^{r_{\text{new}} s G_{\text{key}}}} = M. \tag{7}$$

D. Delivering Result to Control Server From Smart Meter

The smart meter sm_i encrypts the execution result with the control server's RSA PK and sends it back

$$sm_i \rightarrow Control\ Server{:}\ R = \{Result\}RSA^{Control\ Server}_{Public\ Key}.$$

The control server can decrypt it via its RSA private key

$$Result = \{R\}RSA^{Control\ Server}_{Private\ Key}.$$

The additional algorithmic details on P3 can be referred to in our previous preliminary paper on this research [21].

V. PROPOSED APPLICATION

We describe the network communication infrastructure upon which we enable our protocol and the AAC application system together with a case study. (The readers can refer to our previous work [21] for the discussions on why we choose the ABE scheme rather than other solutions such as the symmetric key encryption with pairwise key scheme [32]. Due to the space limit, we will not repeat the details.)

A. Communication Between Publisher and Subscriber

Integration of the ABE encryption primitives and the multicast system is a keystone in P3. In Figs. 2 and 3, we illustrate how control server (publisher) and smart meter (subscriber) send and receive control messages, respectively. They are the basic operations in P3, which enable the multicast sender (control server) to send the messages and make them accessible to its intended recipients (smart meters).

The network communication architecture of P3 and the client–server architecture of P3 are depicted in Figs. 4 and 5, respectively. Fig. 4 illustrates how the ABE encryption system cooperates with the smart grid's multicasting. First, the control server retrieves the access policy associated with the control commands, for instance, {"ZIP = 12345"}. Then, the control messages are bound with the hash result of each clause in the access policy, based on which the control server (publisher) can encrypt messages and the smart meters can decrypt them. After encrypting the commands and calculating the hash results for the access policy, our system binds the multicast address with the outgoing multicast messages. It should be noticed that the multicast address is predefined by the smart grid system rather than P3. The smart meters (subscribers) should bind the same multicast address first and then repeat listening on the arrival of incoming control messages.

B. AAC Application System

In this section, we describe the AAC application system with privacy preserving services that we have designed and developed by utilizing P3 as its cornerstone. It is not only a practical application deployed in a simulated smart grid environment but also a concrete example demonstrating our P3's feasibility. We will focus on its three fundamental subsystems: 1) *Attribute Management* subsystem; 2) *Input* subsystem; and 3) *Encode* and *Decode* subsystems. Then, a brief description of its architecture and a case study follows.

1) Attribute Management Subsystem: We use the *address system* aforementioned as a concrete example to demonstrate how our AAC application is efficiently managed and how the access policy associated with the AAC commands are handled.

2) Input Subsystem: The input subsystem generates and manages requests for the purpose of AAC. There are two sets of input sources: *1) Manual*. Requests can be released by authorized electricity customers via smart phones, web services, and command line applications. A customer, for instance, sends messages via a smartphone to request a service, e.g., turning OFF the A/C at his home. Smartphones, remote access, or web services used here are for command input purpose. Their security can be guaranteed by telecommunication services,

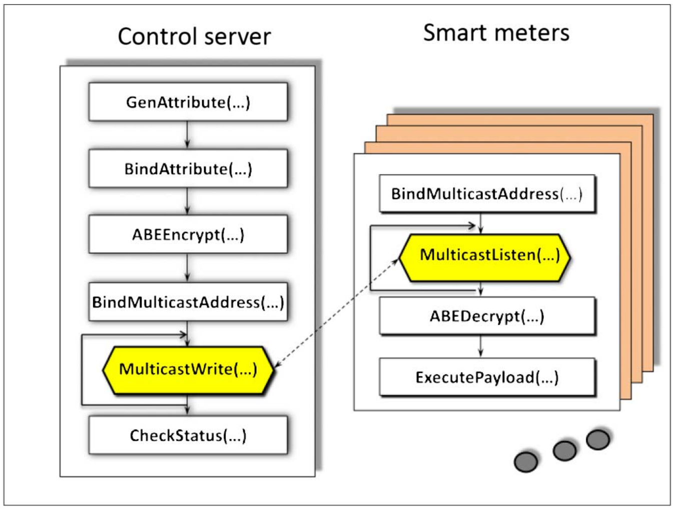

Fig. 4. Network communication architecture for P3.

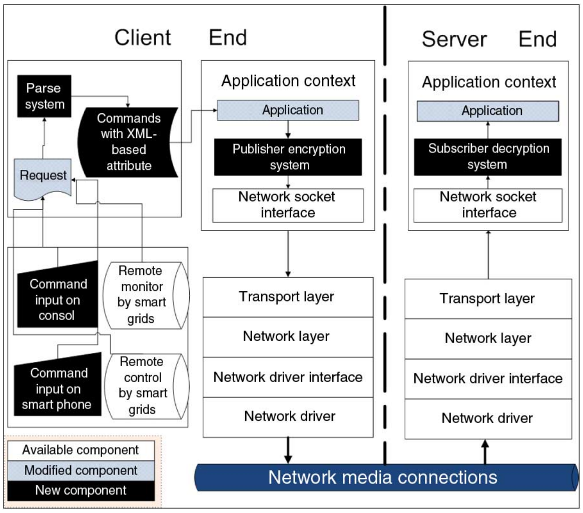

Fig. 5. Client–server architecture for P3.

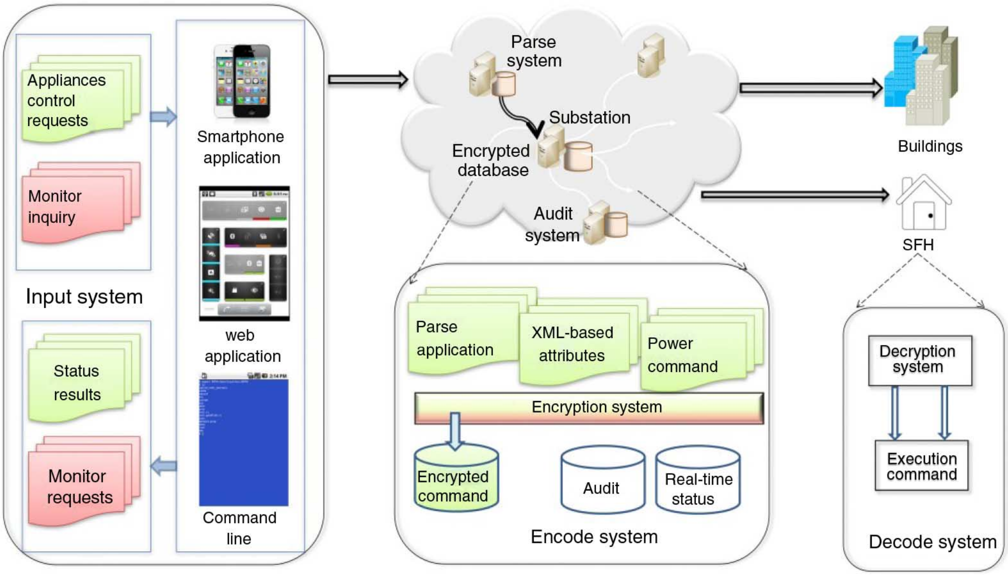

Fig. 6. Case study: AAC system with privacy preservation property.

web/mobile security, or security protocols which are mature and existent in the market. Providing security for them is out of the scope of this paper. 2) *Automatic*. The vast majority of requests are executed by the DR program such as direct load control applications hosted by utility companies.

3) Encode and Decode Subsystems: The *Encode* and *Decode* subsystems process the requests by carrying out three operations in sequence: *1) Parsing*; *2) XML-based Transformation*; and *3) Encryption and Decryption*. The third one is almost the same as P3. The only additional difference is that the control center (e.g., control server) should also validate requests' authorization and verify their authentication. (These aspects will not be further discussed since they are out of this paper's scope.)

4) AAC Architecture: In Fig. 5, a client–server architecture of P3 is presented. It illuminates our detailed architectural design and shows how our P3 can be accommodated in a real smart grid setup. Note that, in the architecture, all *Input* means can send commands in different formats to the *Request* component. The *Request* component, in turn, forwards the received data entry to *Parse* component. The *Parse* component translates data into *XML-based* commands. After that, these commands are encrypted and ciphertext will be multicasted so that the corresponding smart meters can receive and decrypt them.

5) Case Study—Cooperation Among Subsystems: Here, as illustrated in Fig. 6, we demonstrate how the three subsystems cooperate with each other to accomplish the privacy protection task. Smart phone applications, e.g., send out messages listed below to turn OFF an A/C

"$* \mathbf{100} * \mathbf{12345}$ ***Main Street ZIP XYZ, Noname city*** $* \mathbf{3} * \mathbf{2}$"

where "**100**" means the AAC service, "**12345** ***Main Street, ZIP XYZ, Noname city***" stands for the address, "**3**" means the A/C, and "**2**" represents the shutdown command. Alternatively, an e-mail can be sent with the similar format. Or, a web service can be designed which requests customers fill out a form with the same parameters. No matter how many formats utilized in the *Input* subsystem, they all can be parsed into a standardized format, *XML*-based entries. After that, the *Encryption* component encrypts the standardized AAC commands. The control server multicasts ciphertext. After receiving the ciphertext, smart meters invoke the *Decryption* component to decrypt it if the attributes match.

VI. Experimental Results and Security Analysis

A. Evaluation

The performance of the *Encode* component at the client end and the *Decode* component at the server end dominate that of P3. They are the most critical and time-consuming parts in our AAC application. The execution times of the *Input* component (using smartphone, web application, or command line) and the *Parsing* component are trivial. We will not emphasize on their performance here.

Table II evaluates and compares the number of operations for each component in different ABE schemes. Those operations include the exponentiation (E), elliptic curve pairing (P), symmetric encryption/decryption (C), and hashing operations (H). The exponentiation operation is computational expensive, the hashing is light and the pairing and symmetric encryption/decryption are in between. The number of cryptographic operations of ABE encryption/decryption is related to the number of leaf nodes in the access tree of ABE scheme as well as the number of smart meters. The number of cryptographic operations of ABE key update is related to the number of smart meters joining (J)/leave (L) in a rekeying interval (e.g., ranging from one to a few hours) as well as the number of the attributes associated with a smart meter sm.

Previous ABE schemes [6], [18] which utilize one more attribute, namely, the expiration date/time, to expire, validate, or update the ABE key demand and two more exponentiations for each ABE operation. This takes extra times—as observed in our experiments. Thus, as described in Table II, previous ABE scheme demands the most expensive ABE encryption, ABE key

TABLE II
PERFORMANCE EVALUATION: COMPARISON OF ABE COMPONENTS IN P3 WITH OTHERS

Component		*Cost of original ABE Scheme* [6]		*Cost of ABE rekey via group key* [12]		*Cost of our ABE (P3)*	
		Computation	*Communication*	*Computation*	*Communication*	*Computation*	*Communication*
ABE Key	*Encryption*	$2E(1+\lvert\mathrm{AT}\rvert)$	$2(1+\lvert\mathrm{AT}\rvert)\lvert g\rvert$	$2E\lvert\mathrm{AT}\rvert$	$2\lvert\mathrm{AT}\rvert\cdot\lvert g\rvert$	$2E\lvert\mathrm{AT}\rvert$	$2\lvert\mathrm{AT}\rvert\cdot\lvert g\rvert$
	Decry.	$\leq(2P(1+\lvert\mathrm{AT}\rvert)$ $+E\cdot H\cdot\lvert S\rvert)$	$\lvert M\rvert$	$\leq(2P\cdot\lvert\mathrm{AT}\rvert$ $+E\cdot H\cdot\lvert S\rvert)$	$\lvert M\rvert$	$\leq(2P\cdot\lvert\mathrm{AT}\rvert$ $+E\cdot H\cdot\lvert S\rvert)$	$\lvert M\rvert$
ABE Key Update	*Generation*	$2E\cdot(\lvert\mathrm{A_{sm}}\rvert+1)$	$2(1+\lvert\mathrm{A_{sm}}\rvert)\lvert g\rvert$	$2E\cdot\lvert\mathrm{A_{sm}}\rvert$	$2\lvert\mathrm{A_{sm}}\rvert\cdot\lvert g\rvert$	$2E\cdot\lvert\mathrm{A_{sm}}\rvert$	$2\lvert\mathrm{A_{sm}}\rvert\cdot\lvert g\rvert$
	Update	$2(J+L)E\lvert\mathrm{A_{sm}}\rvert$	$2(J+L)\cdot$ $(1+\lvert\mathrm{A_{sm}}\rvert)\lvert g\rvert$	$\leq(\mathrm{F_{H\&K,ABE}}$ $+\mathrm{F_{H\&K,GpKey}})$	$2(J+L)\cdot(\lvert\mathrm{A_{sm}}\rvert\cdot$ $\lvert g\rvert+\sum_{t=1}^{\lvert A\rvert}\log_d T_t\,\lvert G\rvert\cdot\lvert S\rvert)$	$\leq(\mathrm{F_{P3,ABE}}$ $+\mathrm{F_{P3,RSA}})$	$\leq 2\lvert\mathrm{A_{sm}}\rvert$ $\cdot\lvert g\rvert$

E, exponentiation; P, pairing; H, hashing; $g \in \mathbb{G}$, cyclic additive group; C, symmetric encryption; sm, smart meter; $|J|$, number of new-join sm; $|L|$, number of leaving sm; S, set of all sm; $|S|$, number of all sm; $|M|$, length of plaintext; $|G|$, key length of group key G; $|A_{\mathrm{sm}}|$, number of attributes given to sm; $|\mathrm{AT}|$, number of leaves in ciphertext access tree (AT); $|A|$, number of attribute in P3; d, degree of group key tree; $|T_t|$, number of sm association with an attribute t; Refer $F_{\mathrm{H\&K,ABE}}$, $F_{\mathrm{H\&K,GpKey}}$, $F_{\mathrm{p3,ABE}}$, and $F_{\mathrm{P3,RSA}}$ in (8)–(11).

generation, and ABE key update among all three typical ABE schemes. It also invokes the same amount of computational operations for ABE decryption as the other two ABE schemes.

An ABE rekeying scheme in Hur and Kang [12] (denoted as H&K below) deploys the group key agreement for each ABE attribute to update each ABE key. Once there is a smart meter's membership change, each impacted attribute should be updated. As illustrated in (8), in a rekeying interval with J smart meter joining and L smart meter leaving, $(J+L)$ times rekeying computational costs are required. H&K scheme's computation cost including both the ABE rekeying and the group rekeying is listed in (8) and (9). On the other hand, our ABE rekeying scheme in P3 only updates the impacted attributes at the end of the rekeying interval. Therefore, its computational cost in terms of ABE rekeying is significantly reduced. Its computational cost including the ABE rekeying and the RSA PK encryption and the RSA PK decryption is evaluated in (10) and (11) (refer variables/symbols of (8) and (9) in the footnote of Table II)

$$F_{\text{H\&K, ABE}} = (J+L)\left(\sum_{\forall \mathrm{sm}\in S,\ \mathrm{sm}=1}^{|S|} |A_{\mathrm{sm}}|\right) E \tag{8}$$

$$F_{\text{H\& K, GpKey}} = (J+L)\left(\sum_{\forall \mathrm{sm}\in S,\ t=1}^{|A_{\mathrm{sm}}|} d(\log_d T_t)\right) |S|\cdot C \tag{9}$$

$$F_{\text{P3,ABE}} = 2\left(\sum_{\forall \mathrm{sm}\in S,\mathrm{sm}=1}^{|S|} |A_{\mathrm{sm}}|\right) E \tag{10}$$

$$F_{\text{P3,RSA}} = 2\left(\sum_{\forall \mathrm{sm}\in S,\mathrm{sm}=1}^{|S|} |A_{\mathrm{sm}}|\right) E. \tag{11}$$

Comparing (8) with (10), we find that if the rekeying interval is long (e.g., a few hours), $(J+L)$, the number of joining and leave smart meters may possibly be significantly larger than 2, a constant value. We can get the similar observation when (9) is compared with (11). Thus, our ABE scheme demonstrates efficiency as compared with H&K SOLUTION.

1) Cost for Group Rekeying: We notice that both the H&K and our ABE variant utilize the group key to update ABE keys which introduce extra cost. Their computational cost and communication overhead are briefly analyzed and compared. For detailed performance evaluation, refer to [12], [22], and [26].

a) Our ABE variant with method (a) (random value): Since the (a) method described in Section IV-B.3 only generates one random value (G_{key}), its cost almost stays the same and (10) and (11) can be used to evaluate its key management performance. Meanwhile, its communication overhead is N times' unicast, where N is the group size. The computational cost to distribute the new ABE secret keySK is as follows:

$$Comp(N,J,L) = (N+J-L)(E_{\mathrm{RSA}}+D_{\mathrm{RSA}}). \tag{12}$$

b) Our ABE variant with method (b) (periodic rekeying): The (b) method described in Section IV-B.3 constructs and maintains the key tree, the cost of which is evaluated in (13), shown in the next page (with J join and L leave in one rekeying interval) where d is the key tree degree; N is the group size; and $t = \log_d N$ is the height of key tree.

The communication overhead is t times multicast.

c) H & K (individual rekeying):

$$Comp(N,d,J,L) = (dL+2J)\log_d N - L. \tag{14}$$

The communication overhead is $|A|$ times' multicast, where $|A|$ is the number of attributes.

According to the aforementioned evaluation and analyses, we conclude that our approach demonstrates the scalability and is thus more efficient. However, our scheme introduces vulnerability window, although its actual security impact is very minimal in practical smart grid setups.

B. Experiment Results

We implement the *Encode* component and the *Decode* component based on Pairing-Based Cryptography (PBC) library [29] built on the GNU Multiple Precision (GMP) arithmetic library [1]. GMP library provides arbitrary precision arithmetic application programming interfaces (APIs) which are invoked by PBC to support 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 Miyaji, Nakabayashi, and Takano (MNT)

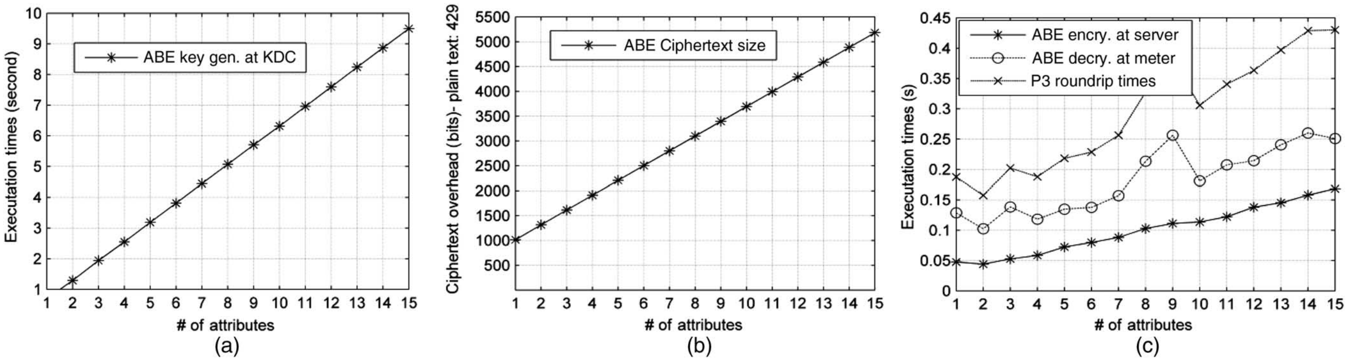

Fig. 7. Experimental test results of ABE systems. (a) ABE key generation on a KDC. (b) ABE ciphertext size. (c) Encryption, decryption, and roundtrip time of P3.

elliptic curve to implement the ABE system. MNT elliptic curve of embedding degree 6 with order 160 bits length and base field order 512 bits length were utilized in P3. We collected 10 times' (randomly selected number) executions of ABE operations, the average of which are depicted at Fig. 7(a)–(c), including (a) ABE key generation on a KDC, (b) ABE ciphertext size when the plaintext size is 429 bits, and (c) Encryption, Decryption, and the roundtrip time of the P3 to process a control 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 was ranging from 1 to 15 (the maximum in practical case). As executing unauthorized 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, Disk—7.9 GB; smart meter: memory—4 MB; CPU—33 MHz, which is the same configuration of a typical real-world smart meter CPU. Unlike experimental configurations in [21], we customize the Ubuntu Operating System in VirtualBox in such a way that only the command line components (e.g., text editors, g++ and gcc, socket functionality, and secure shell (SSH) client and server) are deployed and other packages (audio player, media players, and other GUI applications) are removed. Then, the experimental result shows the much more efficient performance than that in [21].

In Fig. 7, we illuminate the schemes' performance when executing them on the platform mentioned in the figure's caption. The average values of experiment results (the execution is repeated 10 times) above are demonstrated, in which, the ABE encryption at a control server and the ABE decryption at a smart meter executes less than 170 and 260 ms, respectively, when the number of attributes is 15 or less. The roundtrip execution time for P3 takes less than 440 ms when the number of attributes is 15 or less. We also notice that when the number of attributes is 8 or 9, the performance of ABE decryption algorithm is worse than expected. We repeated the experiments for a lot of times and realized that the performance is inconsistent. Sometimes, it is as good as expected but it is not as always. The nonideal-performance result is used in this paper since it may capture the attention from other researchers to dig deeper in future. The ABE communication overhead for P3 is illustrated in Fig. 7(b). It shows the bit sizes of ciphertext transmitted in P3 with attribute numbers ranging from 1 to 15. Though communication overhead is still affordable in the smart grid, its reduction is highly demanded.

Other critical schemes are listed in Table III. The average execution times of ABE setup at the trusted KDC is less than

$$Comp(N,d,J,L) = \begin{cases} d\sum_{l=0}^{t-1} d^l \left(1 - \dfrac{\binom{N-N/d^l}{L}}{\binom{N}{L}}\right), & \text{if } J = L \\ d\sum_{l=0}^{t-1} d^l \left(1 - \dfrac{\binom{N-\frac{N}{d^l}}{L}}{\binom{N}{L}} - \dfrac{\binom{L-J}{N/d^l}}{\binom{N}{N/d^l}}\right), & \text{if } J < L[-0.5pc] \\ d\sum_{l=0}^{t-1} d^l \left(1 - \dfrac{\binom{N-N/d^l}{L}}{\binom{N}{L}}\right) + \left\lceil \frac{d(J-L)}{d-1} \right\rceil, & \text{if } J > L \end{cases} \quad (13)$$

TABLE III
EXECUTION TIMES OF CRYPTOGRAPHIC COMPONENTS

Items	Host	Times (ms)
ABE setup	Trusted KDC	26.450
RSA encrypt	Smart meter	8.096
RSA decrypt	Control server	2.952

27 ms, the RSA encryption at the smart meter is less than 9 ms, and the RSA decryption at the control server is less than 3 ms.

We conclude that they are all sufficiently efficient to be utilized in the AAC system.

C. Security Analysis

We critically examine our system based on generic bilinear group model [5], etc. We argue that it meets the data privacy, namely, distinguishability under chosen-plaintext attack (CPA) and adaptive chosen-ciphertext attacks (CCAs) as no efficient adversary with any reasonable probability can break P3. Without direct access to appliance control commands and their associated access policies, privacy attacks cannot succeed. Full proof can be found in the Appendix.

The security of our ABE scheme's backward secrecy and forward secrecy is based upon the security assumptions of the decisional bilinear Diffie–Hellman (D-BDH). Regarding backward secrecy, when one or a few new smart meters are added, the corresponding secret key SK' is generated based on the ABE secret key generation algorithm with the input of r_{new} as well as a few r_j^{new} corresponding to the new smart meter's attributes. Note that when the new smart meters try to compromise the backward secrecy, they need guess the existing smart meters' r' and their r'_j which are randomly generated at prior rekeying windows. It is not computationally feasible to deduce them since the D-BDH assumption holds. Furthermore, the new secret key SK' will be forwarded after it is encrypted by RSA keys. Therefore, the existing smart meter's secret key, SK cannot be leaked to any new smart meters. Following the similar arguments, the leaving smart meters cannot guess the newly generated/calculated G_{key}. According to our decryption algorithm, G_{key} is necessary. Consequently, our ABE scheme guarantees both forward secrecy and backward secrecy, the formal proof of which can be easily achieved via extending the model utilized in Theorems 1 and 2. Refer to the Appendix for detailed security analyses.

Theorem 1: Suppose the D-BDH assumption holds. There is no polynomial-time adversary A that can break semantic security of ABE components in P3 system by CPA.

Theorem 2: Suppose the D-BDH assumption holds. There is no polynomial-time adversary A that can break semantic security of ABE components in P3 system by CCA.

Theorem 3: Suppose the Decisional Diffie–Hellman (DDH) assumption holds. There is no polynomial-time adversary A that can break semantic security of RSA PK components in P3 system by CPA.

Theorem 4: Suppose the DDH assumption holds. There is no polynomial-time adversary A that can break semantic security of RSA PK components in P3 by CCA.

D. Analyses and Future Works

Improving the performance of ABE key management and deploying the P3 in AAC applications in smart grid are described in this paper. However, this paper still leaves open some challenging issues which can inspire future research efforts.

1) *Transformation of Address Management System*: In smart grids, a set of policies is established to control the power distribution. In this paper, we use *streets*, *ZIP*, and *cities*, as examples because it is easier for end customers to remotely control their appliances. But, some utilities may use electrical terms such as "*district #*," "*sub-district #*," "*substation #*," and "*feeder #*." 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. However, an efficient and feasible transformation subsystem is requested as our future work.
2) *Authorization for Installation or Expiration of Smart Meters*: Before a new smart meter is legally installed at the residence, its authorization need to be verified. While a smart meter is expired, the ABE keys, group keys, and temporary keys that it contains should be completely deleted in such a way that the adversaries cannot take advantage of it. Furthermore, how to accommodate the legacy smart devices in our P3 system is also a thought-provoking issue.
3) *Efficient Privacy Preservation in Other Parts*: Our P3 covers privacy of the communication between the utilities and the smart meters. Our previous research [19] exactly conceals the privacy of the communication between the smart meter and the appliances. In [19], an efficient hybrid group key scheme plus symmetric encryption algorithm is used to hide the privacy. However, a more efficient and flexible solution, e.g., a tailored pairwise key agreement is demanded since it requests less memory storage. Furthermore, based on our assumption, this paper does not consider the privacy leakages at the control server end. However, a number of popular third-party applications, e.g., a data-mining application deployed at the control server end to process the real-world smart grid data may breach the privacy protection at the utility. That will be our future research.
4) *Theoretical Framework for Privacy*: The privacy can also be leaked by analyzing the end customers' electricity demand profiling. Though it is beyond the scope of this paper, future research can study possible solutions by utilizing the theoretical framework proposed in [14].

VII. RELATED WORKS

Preserving privacy for AAC applications or DR in the smart grid has not been well explored before. However, privacy preservation approaches for the smart grid, in general, have been studied by means ranging from *privacy theory framework*, *battery* [15], [30], *identification (ID) anonymization* [10], *disturbance* [24], *cryptographic schemes* [11], [12], [17], [36]. They were mainly designed to hide power consumption data aggregated from smart meters and sent to the utilities. Furthermore, in [27], a set of possible privacy attacks that discloses occupant's activities in-home through the usage of power consumption data in DR were demonstrated. Unlike previous research which mainly focuses on hiding power consumption data sent from smart meters to utilities, our paper guards not only the multicast command messages forwarded from utilities to smart meters, but also the meter events/execution results of smart meters or appliance which are sent back to the utilities.

A. Battery

Some privacy protections use rechargeable battery. In [35] , to control the energy flow within a home, Rajagopalan *et al.* utilize the electric power routing by running partial power consumption demands off a rechargeable battery rather than off the power grid directly. It offsets the power usage activity and moderates load signatures' effect. Kalogridis *et al.* [15] proposed the *ElecPrivacy* system to detect ongoing or upcoming privacy threats, reconfigure the power routing and eventually mask load signature for appliances. McLaughlin *et al.* [30] proposed the NonIntrusive Load Leveling (NILL), a new class of algorithms to mask the appliance's power usage signature. However, rechargeable batteries cost around 1000 [30]. They also require installment and maintenance expenses. Furthermore, nowadays, smart appliances such as A/Cs, dryers, and dishwashers. can directly communicate with utility operators. Hence, installing rechargeable batteries cannot totally mask all the appliances' load signatures.

B. Cryptographic Schemes

Li *et al.* [23] focused on smart metering data aggregation protection in which all messages were encrypted via the homomorphic encryption algorithm. Garcia and Jacobs [11] proposed a privacy-friendly protocol by using homomorphic (Paillier) encryption and additive secret sharing. Rial and Danezis [36] used zero knowledge proofs and commitments to preserve smart meters' privacy. In [17], Kursawe *et al.* presented four different protocols based on Diffie–Hellman key-exchange to protect privacy of metering data aggregation. However, to our best knowledge, so far no cryptographic solutions have been proposed to deal with privacy leakage in appliance control applications yet.

C. Anonymity

Efthymiou and Kalogridis [10] proposed a trusted key escrow service to anonymize frequent readings with pseudonymous IDs rather than unique identifiers along with randomized time intervals. Nevertheless, anonymity approaches masking customers' identity cannot preserve customers' behavior once the escrow service is compromised.

D. Disturbance

Li *et al.* [24] proposed a compressed meter reading approach that enhances its privacy through the use of random sequences. But the method unrealistically assumed that its access points will never be compromised.

E. ABE Key Revocation

The original ABE system [6] or its variants [18] revoke expired ABE keys via expiry date/time. But they cannot handle newly joined smart meters in such a way that *backward privacy* is efficiently ensured. The attribute revocation scheme is proposed in [12] to satisfy the ABE rekey requirement of military networks. The group key management scheme (e.g., [12]) is utilized to generate the attribute group key k_{λ_y} for each attribute in ABE. The updated ABE secret key can be delivered to each affected smart meter with the encryption of k_{λ_y}. However, this requires each smart meter to store additional $\log n$ KEKs, which are auxiliary keys to facilitate rekeying operations. So, it is not efficient in terms of memory and hence not suitable for smart meters with limited memory. Moreover, smart meters cannot process the group key management operation too frequently due to restricted processing capabilities. Ostrovsky *et al.* proposed the immediate key revocation scheme through the usage of negative clauses [34]. The revoked smart meters' identification is added under the AND gate conjunctively with the negation. But, still, its performance is not efficient especially for smart meters. CP-ABE and KP-ABE are introduced into Information Centric Networking and a case study of the smart city has been conducted [13].

VIII. Conclusion and Future Works

The smart grid provides the strongest example of a current IoT deployment. AAC applications present much convenience to customers in the smart grid. However, AAC commands can easily be mined to expose customers' privacy such as absences, appliance ownerships, and daily activity models. To protect the customers' sensitive information, we propose the P3 through the use of our adapted ABE variant scheme coupled with a suitable key management scheme and RSA algorithm. Based on our P3, we further design and develop an AAC system with the privacy preservation service. Our experimental results show that the computational cost and the delay incurred by the cryptographic approaches are significantly light and especially suitable to be deployed in resource-limited smart meters.

This paper leaves a few open problems which are of outstanding challenges. It is the objective of our future research to further optimize the performance for ABE key revocation, minimize its vulnerable window, authorize the legal activities of smart meters' memberships (e.g., installation or expiration), and cover the privacy for both the control server end and the appliance end in the AAC application. In addition, proposing more powerful schemes for stronger privacy preservation services such as *unlinkability* for access policies, adapting multiauthority ABE into our P3 and prove our P3 in theoretical framework are also in our future research effort.

Appendix

A. Privacy Preservation Analysis

We know that if the ciphertext generated by the ABE scheme or the RSA PK system is provably secure, the ciphertext delivered on communication channels of P3 system can provide data privacy. Thus, in this section, we prove that ABE and RSA components in P3 system are secured sufficient.

We first describe the D-BDH assumption which is the cornerstone of the P3's semantic security we are going to prove. Second, we prove the security of ABE components utilized in P3 then it follows the security of RSA PK encryption component in P3.

1) Assumptions:

a) D-BDH assumption: Let $a, b, c, z \xleftarrow{R} \mathbb{Z}_P$. There are two tuples: $(A = g^a,\ B = g^b,\ C = g^c,\ \hat{\mathrm{e}}(g, g)^{abc})$ as well as $(A = g^a,\ B = g^b,\ C = g^c,\ \hat{\mathrm{e}}(g, g)^z)$. 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

$$\begin{aligned} \boldsymbol{Adv}_{\mathcal{A}} = \big|&\Pr\left[\mathcal{A}\left(A, B, C, \hat{\mathrm{e}}(g, g)^{abc}\right) = 0\right] \\ &- \Pr\left[\mathcal{A}(A, B, C, \hat{\mathrm{e}}(g, g)^z) = 0\right]\big|. \end{aligned} \tag{15}$$

2) Data Privacy in ABE Component of P3:

Definition 1 (ABE-CPA). Let $\mathcal{P} = \mathcal{S}, \mathcal{G}, \mathcal{E}, \mathcal{D}$ be the ABE system in P3 which encrypts/decrypts utility messages M in transmission. $\mathcal{S}$ stands for ABE setup, $\mathcal{G}$ 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 chosen plaintext attacks launched by all polynomial time complexity adversaries $\mathcal{A}$ if $\mathcal{A}$'s $\boldsymbol{Adv}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-b}(k)$ is negligible. The security model we are going to use follows the experiment listed below.

Experiment $\boldsymbol{Exp}_{\mathcal{PA},\mathcal{A}}^{\mathrm{ABE-CPA}-b}(k)$

$$\begin{aligned} &(\mathrm{PK,MSK}) \xleftarrow{R} \mathcal{S}(k); \\ &\mathrm{SK} \xleftarrow{R} \mathcal{G}(\mathrm{MSK}); \\ &M_0 \xleftarrow{R} \{0,1\}^*; \quad M_1 \xleftarrow{R} \{0,1\}^*; \\ &\mathrm{CT}_b \leftarrow \mathcal{E}(\mathrm{PK}, M_b); \\ &M_b \leftarrow \mathcal{A}(\mathrm{find}, \mathrm{CT}_b, M_0, M_1); \\ &\mathrm{return}: \quad g \leftarrow \mathcal{A}(\mathrm{guess}, \mathrm{CT}_b). \end{aligned}$$

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 PKs which can be used by $\mathcal{A}$ to generate any number of ciphertexts within polynomical 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 adversaries $\mathcal{A}$ 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

$$\begin{aligned} \boldsymbol{Adv}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-b}(k) &= \Pr[\boldsymbol{Exp}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-0}(k) = 0] \\ &\quad - \Pr[\boldsymbol{Exp}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-1}(k) = 0] \\ &= 2 \cdot \Pr[\boldsymbol{Exp}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-0}(k) = 0] - 1. \end{aligned} \tag{16}$$

Theorem 1: Suppose the D-BDH assumption holds. There is no polynomial-time adversary $\mathcal{A}$ that can break semantic security of ABE components in P3 system by CPA.

Proof: Suppose we have an adversary $\mathcal{A}$ with negligible advantage. $\epsilon = \boldsymbol{Adv}_{\mathcal{P},\mathcal{A}}^{\mathrm{ABE-CPA}-b}(\cdot)$. which can break ABE components in P3 system. A simulator $\boldsymbol{\mathcal{B}}$ which plays the decisional BDH game with advantage ϵ processes in the following way.

Init: Let the adversary $\mathcal{A}$ randomly chooses the set of challenge access structure, namely $\boldsymbol{\mathcal{T}}^*$ which will be challenged upon.

Setup: The simulator $\boldsymbol{\mathcal{B}}$ first randomly generates two credentials $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p$. After then, $\boldsymbol{\mathcal{B}}$ sends adversary $\mathcal{A}$ the following PKs:

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

We then showcase how the simulator $\boldsymbol{\mathcal{B}}$ programs each node $y \in Y$, where Y are set of leaf nodes in the tree $\mathcal{T}^*$.

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

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

$$D = g^{\frac{\alpha + r^{(j)}}{\beta} G_{\mathrm{key}}} \tag{17}$$

and $\forall j \in s_j$: $\{D_i = g^{r^{(j)} + t_i r_i^{(j)}};\ D'_i = g^{r_i^{(j)}}\}$.

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

Challenge: The adversary $\mathcal{A}$ submits two challenge message M_0 and M_1 and the access tree $\mathcal{T}^*$ to the simulator $\boldsymbol{\mathcal{B}}$. The simulator $\boldsymbol{\mathcal{B}}$ needs to compute one of $M_0\hat{e}(g, g)^{G_{\mathrm{key}}\alpha s}$ and $M_1\hat{e}(g, g)^{G_{\mathrm{key}}\alpha s}$, where $\alpha, s \xleftarrow{R} \mathbb{Z}_p$. Here, we consider a modified game, where $\tilde{C}$ is calculated by either $\hat{e}(g, g)^{G_{\mathrm{key}}\alpha s}$ or $\hat{e}(g, g)^{\theta}$ where $\theta \xleftarrow{R} \mathbb{Z}_p$ Therefore, the adversary $\mathcal{A}$ with advantage ϵ for ABE component in P3 can be transformed into a new adversary with the advantage of $\epsilon/2$. To simplify, we will use the modified game from now on. Based on the notions aforementioned, the simulator $\boldsymbol{\mathcal{B}}$ processes the following: First, $s \xleftarrow{R} \mathbb{Z}_p$. Second, the linear secret sharing scheme associated with access tree is used to construct share λ_i of s for all relevant attributes i. Third, the simulator $\boldsymbol{\mathcal{B}}$ chooses $\theta \xleftarrow{R} \mathbb{Z}_p$. Fourth, the simulator. $\boldsymbol{\mathcal{B}}$. flips a fair coin $\mu \in \{0, 1\}$ which is beyond the awareness of adversary $\mathcal{A}$. At last, accomplish the following encryption:

$$\begin{aligned} &\tilde{C} = M_{\mu} e(g, g)^{\theta} \quad C = h^s \\ &\qquad\qquad \forall i \in Y\colon \{C_i = g^{\lambda_i}; C'_i = g^{t_i \lambda_i};\}. \end{aligned}$$

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

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

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

$$\Pr[B(\mathrm{PK}, D, D_i, T = e(g, g)^{\theta}) = 0] = {}^1\!/_2 + \mathrm{Adv}_{\mathcal{A}}. \quad (18)$$

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

$$\Pr[B(\mathrm{PK}, D, D_i, T = \mathit{Random}) = 0] = {}^1\!/_2. \quad (19)$$

Consequently, the simulator $\boldsymbol{\mathcal{B}}$ plays the decisional BDH game with nonnegligible advantage. ■

Theorem 2: Suppose the D-BDH assumption holds. There is no polynomial-time adversary $\mathcal{A}$ that can break semantic security of ABE components in P3 system by CCA.

Proof: The model utilized in Theorem 1 can easily be extended to prove CCA by allowing random oracle techniques for decryption in Phases 1 and 2. ■

3) *Data Privacy in RSA PK Component of P3:*

a) Decisional 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 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

$$\boldsymbol{Adv}_{\mathcal{B}} = \left|\Pr\left[\mathcal{B}(A, B, g^{ab}) = 0\right] - \Pr\left[\mathcal{B}(A, B, g^{c}) = 0\right]\right|. \quad (20)$$

Definition 2 (RSA-CPA): Let $\mathcal{P} = (\mathcal{S}, \mathcal{G}, \mathcal{E}, \mathcal{D})$ be the RSA PK system in P3 which encrypts/decrypts metering messages M in transmission from smart meters to the control server. $\mathcal{S}$ stands for RSA setup, $\mathcal{G}$ 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 3: Suppose the DDH assumption holds. There is no polynomial-time adversary $\mathcal{A}$ that can break semantic security of RSA PK components in P3 system by CPA.

Proof: The model utilized in Theorem 1 can easily be reused to prove RSA-CPA. ■

Theorem 4: Suppose the DDH assumption holds. There is no polynomial-time adversary $\mathcal{A}$ that can break semantic security of RSA PK components in P3 by CCA.

Proof: The model utilized in Theorem 1 can easily be extended to prove CCA by allowing random oracle techniques for decryption in Phases 1 and 2. ■

References

[1] The GNU MP Bignum Library [Online]. Available: http://gmplib.org/, accessed on Feb. 9, 2014.

[2] (2011). *Control Your House Lights (and more) with Your iPhone* [Online]. Available: http://cybernetnews.com/control-lights-with-your-iphone/, accessed on Feb. 9, 2014.

[3] (2004). *Control Your Appliances Over The Internet* [Online]. Available: http://www.popularmechanics.com/technology/gadgets/1279916, accessed on Feb. 9, 2014.

[4] Z. Aung, M. Toukhy, J. Williams, A. Sanchez, and S. Herrero, "Towards accurate electricity load forecasting in smart grids," in *Proc. IARIADBKDA'12*, 2012, pp. 51–57.

[5] P. Barreto, B. Lynn, and M. Scott, "Efficient implementations for pairing-based cryptography," *J. Cryptol.*, vol. 17, pp. 321–334, 2004.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE S&P'07*, 2007, pp. 321–334.

[7] J.-M. Bohli and A. Pashalidis, "Relations among privacy notions," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 4, pp. 362–380, 2011.

[8] D. Boneh and M. Franklin, "Identity-based encryption from Weil pairing," in *Proc. Crypto'01*, 2001, vol. 2139, pp. 213–229.

[9] H. S. Cho, T. Yamazaki, and M. Hahn, "AERO: Extraction of user's activities from electric power consumption data," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 2011–2018, Aug. 2010.

[10] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proc. IEEE SmartGridComm'10*, 2010, pp. 238–243.

[11] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proc. STM'10*, 2010, vol. 6710, pp. 226–238.

[12] J. Hur and K. Kang, "Secure data retrieval for decentralized disruption-tolerant military networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 16–26, Feb. 2014.

[13] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in ICN: Attribute-based encryption and routing," in *Proc. 3rd ACM SIGCOMM Workshop Inf.-Centric Netw. (ICN'13)*, Hong Kong, 2013, pp. 39–40.

[14] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Proc. IEEE SmartGridComm'10*, 2010, pp. 232–237.

[15] G. Kalogridis, R. Cepeda, T. Lewis, S. Denic, and C. Efthymiou, "ElecPrivacy: Evaluating the privacy protection of electricity management algorithms," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 750–758, Dec. 2011.

[16] M. Karwe and J. Strüker, "Maintaining privacy in data rich demand response applications," in *Smart Grid Security*, vol. 7823. Berlin, Germany: Springer-Verlag, 2013, pp. 85–95.

[17] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Proc. PETS'11*, 2011, vol. 6794, pp. 175–191.

[18] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT'11*, 2011, vol. 6632, pp. 568–588.

[19] D. Li, Z. Aung, S. Sampalli, J. Williams, and A. Sanchez, "Privacy preservation scheme for multicast communications in smart buildings of the smart grid," *Smart Grid Renew. Energy*, vol. 4, pp. 313–324, 2013.

[20] D. Li, Z. Aung, J. Williams, and A. Sanchez, "Efficient authentication scheme for data aggregation in smart grid with fault tolerance and fault diagnosis," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT'12)*, 2012, pp. 1–8.

[21] D. Li, Z. Aung, J. Williams, and A. Sanchez, "P3: Privacy preservation protocol for appliance control application," in *Proc. IEEE SmartGridComm'12*, 2012, pp. 294–299.

[22] D. Li and S. Sampalli, "A hybrid group key management protocol for reliable and authenticated rekeying," *Int. J. Netw. Secur.*, vol. 6, pp. 270–281, 2008.

[23] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. IEEE SmartGridComm'10*, 2010, pp. 327–332.

[24] H. Li, R. Mao, L. Lai, and R. C. Qiu, "Compressed meter reading for delay-sensitive and secure load report in smart grid," in *Proc. IEEE SmartGridComm'10*, 2010, pp. 114–119.

[25] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 686–696, Dec. 2011.

[26] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, "Batch rekeying for secure group communication," in *Proc. 10th World Wide Web Conf. (WWW'10)*, Hong Kong, May 2010, pp. 525–534.

[27] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Secur. Privacy*, vol. 8, no. 1, pp. 11–20, Jan./Feb. 2010.

[28] M. A. Lisovich and S. B. Wicker, "Privacy concerns in upcoming residential and commercial demand-response systems," in *Proc. Clemson PS'08*, 2008, pp. 1–8.

[29] B. Lynn. (2006). *The Stanford Pairing Based Crypto Library* [Online]. Available: http://crypto.stanford.edu/pbc/, accessed on Apr. 13, 2013.

[30] S. McLaughlin, P. McDaniel, and W. Aiello, "Protecting consumer privacy from electric load monitoring," in *Proc. ACM CCS'11*, 2011, pp. 87–98.

[31] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel, "Multi-vendor penetration testing in the advanced metering infrastructure," in *Proc. 26th Annu. Comput. Secur. Appl. Conf.*, 2010, pp. 107–116.

[32] A. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1997.

[33] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 99–107, Jun. 2010.

[34] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. ACM CCS'07*, 2007, pp. 195–203.

[35] S. R. Rajagopalan, L. Sankar, S. Mohajer, and H. V. Poor, "Smart meter privacy: A utility-privacy framework," in *Proc. IEEE SmartGridComm'11*, 2011, pp. 190–195.

[36] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proc. ACM WPES'11*, 2011, pp. 49–60.

[37] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 382–390, Jun. 2011.

[38] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the IoT," in *Proc. IEEE ICC'14*, Sydney, N.S.W., Australia, 2014, pp. 1–6.

[39] Y. R. Yang, X. S. Li, X. B. Zhang, and S. S. Lam, "Reliable group rekeying: A performance analysis," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 27–38, 2001.

[40] J. Zhang and C. A. Gunter, "Application-aware secure multicast for power grid communication," in *Proc. IEEE SmartGridComm'10*, 2010, pp. 339–344.

Depeng Li, photograph and biography not available at the time of publication.

Zeyar Aung, photograph and biography not available at the time of publication.

John Wiliams, photograph and biography not available at the time of publication.

Abel Sanchez, photograph and biography not available at the time of publication.