

## Chapter 2

# Equality Query for Auction in Emerging Smart Grid Marketing

Distributed energy resources (DERs), which are characterized by small scale power generation technologies to provide an enhancement of the traditional power system, have been strongly encouraged to be integrated into the smart grid, and numerous trading strategies have recently been proposed to support the energy auction in the emerging smart grid marketing. However, few of them consider the security aspects of energy trading, such as privacy-preservation, bid integrity and pre-filtering ability. In this chapter, we introduce an efficient Searchable Encryption Scheme for Auction (SESA) in emerging smart grid marketing. Specifically, SESA uses a public key encryption with keyword search technique to enable the energy sellers, e.g. DERs, to inquire suitable bids while preserving the privacy of the energy buyers (EBs). Additionally, to facilitate the seller to search for detailed information of the bids, we also propose an extension of SESA to support conjunctive keywords search.

### 2.1 Introduction

Growing demand for electricity, upcoming fossil-fuel shortage and CO<sub>2</sub> emission crises have recently invoked an urgent need in incorporating renewable energy sources into the power grid. Such a trend is commonly known as distributed generation (DG) [1]. In the trend of DG, distributed energy resources have been encouraged to participate in energy marketing to facilitate competition among different energy providers. However, how to negotiate with different energy providers and energy consumers is a challenging issue in DG [2]. In order to address this challenge, smart grid, which is composed of many entities: intelligent electricity distribution devices, advanced sensors, two-way automated metering infrastructure, and specialized computer systems to enhance the operation performance [2], has received significant attention in recent years. Smart grid can accelerate the integration of distributed energy suppliers, DERs and microgrids [3], and thus it

potentially makes power generation, transmission, and distribution be the next big e-business operating mostly under autonomous control [4]. As a result, smart grid has been recognized as an emerging electricity market.

In order to protect users' information privacy and security during the auction process, each buyer should protect their bidding information and let it not be known by the unauthorized users, including the auction server. While at the same time, it enables the sellers to query the auction server about the demanded bids. Although many auction models (e.g. [2, 5, 6]) were established respectively for smart grid energy marketing, few of them takes the privacy or security of the DERs into consideration. Recently, various security vulnerabilities and threats have been studied in the research literatures [7–9].

Lu et al. [10] used a super-increasing sequence to structure multi-dimensional data and encrypt the structured data by the homomorphic paillier cryptosystem technique. Li et al. [11] proposed an efficient demand response scheme to achieve privacy preserving demand aggregation and efficient response. However, since these encryption schemes can not be searched, they are not suitable for auction in smart grid marketing. On the other hand, some of the traditional auction schemes [12, 13] can achieve bidding privacy, but they can not support keyword search or bids filtering.

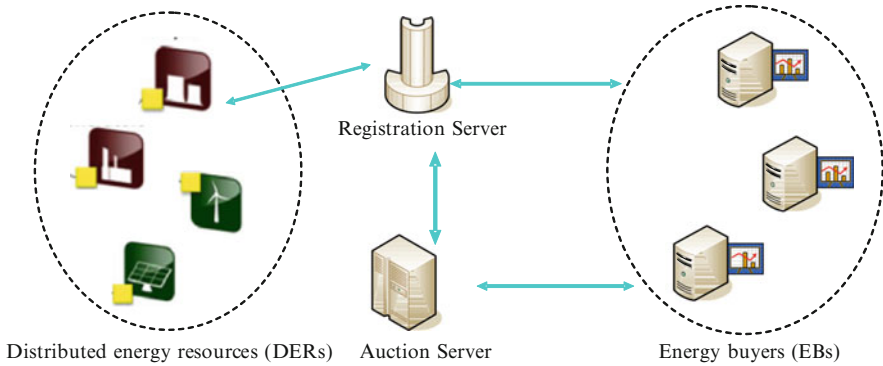
In this chapter, we address the efficient searchable encryption problem for auction in smart grid marketing. This scheme considers both the public key based encryption and keyword search techniques. It can achieve privacy-preservation, searchable ability and bids filtering, as well as other security features including confidentiality, authenticity and integrity. The main content of this chapter are twofold.

1. Firstly, a novel SESA scheme is constructed to achieve searchable encryption, by modifying the proxy re-encryption with keyword search scheme [14]. The security analysis demonstrates that SESA can achieve confidentiality, data and keyword privacy, authenticity and data integrity.
2. Secondly, we construct an extended version of SESA to support conjunctive keywords search. It enables the user to question the auction server more flexibly.

The remainder of this chapter is organized as follows. In Sect. 2.2 we describe the smart grid marketing architecture, security requirement and design goal. Then, we present the SESA scheme and its extension in Sects. 2.3 and 2.4, followed by its security analysis and performance evaluation in Sects. 2.5 and 2.6, respectively. Next, we review the related works in Sect. 2.7. Finally, we draw our summary in Sect. 2.8.

## 2.2 System Model and Design Goal

In this section, we formalize the system model, security requirements, and identify our design goals.



**Fig. 2.1** Smart grid marketing architecture

### 2.2.1 Smart Grid Marketing Architecture

Smart grid marketing refers to a system that enables small producers to generate and sell electricity at the local level. As shown in Fig. 2.1, there are energy sellers (e.g. DERs), energy buyers (e.g. EBs), and auction managers. The auction managers are two servers: a registration server (RS) and an auction server (AS).

- RS:** In an energy marketing, a registration server is used to initiate the system at the beginning of the auction; and when the bidding is finished, it will select the winner according to the criteria of the DERs. The RS is trustworthy and it will send some keywords from the DERs to the auction server to search for their wanted bids. The winner may be selected from these pre-filtered bids.
- AS:** Auction server is used in a continuous sealed-bid auction in which traders submit offers to buy (bid) or offers to sell (ask) at any time during the trading period. The auction server is semitrust and it cannot know the content of the EBs' bids, but it can test if the message has tags like the seller's query.
- DER:** DERs can open the bids by themselves. However, due to the number of distributed bids from EBs may be large, to improve the efficiency, the RS will act as a proxy for the DERs to select the winners.
- EB:** Energy is bought from or sold to the grid depending on the availability, demand, and price of energy. Each energy buyer will send its sealed-bid to the auction server. Due to the large amount of buyers, the bids may be conducted with the competition of others.

### 2.2.2 Security Requirements

We assume that the communication between EBs and server is untrustworthy. That is, various adversaries such as eavesdroppers and tamperers may be present. If a large amount of EBs are competitive to buy a certain type of energy from DERs, it is reasonable to enable the RS to query the AS and select one or group of winners according to the criteria of the DERs.

We define the security requirements for our SESA scheme, and will show the fulfillment of these requirements after presenting the design details.

- *Data privacy*: The data owner can resort to the public key cryptography to encrypt the data before outsourcing, and successfully prevent the unauthorized entities, including the auction server, from prying into the outsourced data.
- *Bid integrity*: The bids information and queries should not be changed by the malicious users or the illegal competitors, i.e., if the competitor  $\mathcal{A}$  maliciously modified the price or other information of  $EB_i$ , it may lead to  $EB_i$  can not be selected by the RS.
- *Keyword privacy*: As users usually prefer to keep their search from being exposed to others, including the auction server, the most important concern is to hide what in their bids and what the RS is inquiring, i.e., the keywords indicated by the corresponding trapdoor. Thus, the trapdoor should be generated in a cryptographic way to protect the query keywords.
- *Trapdoor unforgeability*: DER generates his trapdoor information based on his keyword and secret key. After the AS receiving the trapdoor, it can test this trapdoor with keyword tags. The most important thing is that others (include the AS) can get nothing from the trapdoor, i.e. the AS cannot forge a new trapdoor based on the old ones.

### 2.2.3 Design Goal

To enable searchable encryption for effective utilization of outsourced energy bids under the aforementioned model, our design goal is to develop a searchable encryption scheme for auction in emerging smart grid marketing, and achieve the security of the bids and efficient keyword search as follows.

- The proposed scheme should achieve security as mentioned in the security requirements, i.e., the data privacy, keyword privacy, data integrity and trapdoor unforgeability.
- The proposed scheme should achieve both one keyword and conjunctive keywords search.
- The proposed scheme should achieve the communication and computation efficiency, compared with other searchable encryption schemes.

## 2.3 SESA Scheme

In this section, we show the construction of the efficient Searchable Encryption Scheme for Auction (SESA) in emerging smart grid marketing, which mainly consists of the following four phases: Registration phase, Bidding phase, Pre-filtering phase, and Decision-of-winner phase. For our auction system, we assume that there is a local registration server (RS) which can bootstrap the system. Specifically, in this system initialization phase, given the security parameter  $1^k$ , RS first generates  $(q, g, G_1, G_2, e)$  by running  $\mathcal{Gen}(1^k)$ , where  $q$  is a  $k$ -bit prime number. Let  $G_1$  and  $G_2$  be two cyclic multiplication groups.  $Sig(\mathcal{G}, U, V)$  is an identity based signature scheme [15]. Furthermore, we will need three hash functions  $H_1 : \{0, 1\}^* \rightarrow G_1$ ,  $H_2 : \{0, 1\}^* \rightarrow G_1$ ,  $H_3 : G_2 \rightarrow \{0, 1\}^*$ . RS publishes the system parameters as  $(q, g, G_1, G_2, e, H_1, H_2, H_3)$ .

### 2.3.1 Registration Phase

In order to maintain security of the network against attacks and the fairness among customers and providers, the local RS may control the access of each DER and EB. The energy marketing announces two prices: the price for selling energy and the price for buying energy in the smart grid marketing. The DERs adjust their bidding price after negotiating with the other units based on the grid prices, considering their operational cost and local demands.

In our scheme, there are  $n$  DERs and  $m$  EBs in the energy marketing. For each  $DER_i$  ( $i = 1, \dots, n$ ) and  $EB_j$  ( $j = 1, \dots, m$ ), when they register, the RS picks two random numbers  $x_i, y_i \in \mathbb{Z}_q^*$  and sets  $pd_i = g^{x_i}$ ,  $pb_j = g^{y_j}$ .  $(pd_i, x_i)$  and  $(pb_j, y_j)$  are  $DER_i$ 's and  $EB_j$ 's public/private key pairs, respectively. For each  $EB_j$  ( $j = 1, \dots, m$ ), the RS randomly chooses a master key  $s \in \mathbb{Z}_q^*$  and assigns an ID-based key pair  $(H_1(ID_{EB_j}), H_1^s(ID_{EB_j}))$  to  $DER_i$  for signature, and denotes it as  $(vk_j, ssk_j)$ .

In the energy marketing, the  $DER_i$  will publish its energy information  $m_i = (p_i, GID_i, Ts, Lo_i, Am_i, T_N)$  publicly, where  $p_i$  is the initial price,  $GID_i$  is the identification of the energy,  $Ts$  is the timestamp,  $Lo_i$  is the energy resource location,  $Am_i$  is the amount of the energy and  $T_N$  is the unique serial number of the deposit energy information. The RS will store the information from each  $DER_i$  as a tuple  $(DER_i, m_i)$  in its database. Also,  $EB_j$  will register its personal information  $e_j = (Lo_j, Rep_j, Ty_j, \Delta)$  on the RS, where  $Lo_j$  is its location,  $Rep_j$  is its reputation about its history trades (which also will be verified by the RS, but it's not our paper's focus),  $Ty_j$  is the demanded energy types,  $\Delta$  is the other information of  $EB_j$ . The RS also stores the information from each  $EB_j$  as a tuple  $(EB_j, e_j)$  in its database.

### 2.3.2 Bidding Phase

In order to achieve the nearly real-time energy bidding, each  $EB_j$  will choose its interested energy to bid. The bidding is performed as following steps.

1.  $EB_j$  gets an identity based signature key pair as  $(vk_j, sk_j)$  from RS. The public key is represented as  $A = vk_j$ , and the private key  $sk_j$  is kept secretly.
2.  $EB_j$  selects a random  $r_j \in \mathbb{Z}_q^*$ , and generates a  $bid_j = (EB_j, pr_j, GID_i, Cb_j, Rep_j)$ , where  $pr_j$  is the price of the bid,  $Cb_j$  is the amount of the energy that  $EB_j$  wants to buy,  $Rep_j$  is  $EB_j$ 's reputation. Then,  $EB_j$  computes  $C_j = H_3(e(g, H_2(A)^{r_j})) \oplus bid_j$ .
3. In order to maximize the probability of winning in the auction,  $EB_j$  selects a keyword  $w_j$  to represent his bid (e.g. the reputation or required amount). Next,  $EB_j$  computes a tag on the keyword as  $t_j = e(g, H_1(w_j)^{r_j})$ . Then he computes  $B_j = (g^{x_i})^{r_j}$  and  $F_j = H_3(t_j)$ . He outputs  $C'_j = (B_j, F_j)$ .
4.  $EB_j$  generates a signature  $S_j = Sig_{sk_j}(C_j, C'_j)$ .  $(C_j, C'_j)$  is the signed message.
5.  $EB_j$  sends the encrypted message  $K_j = (GID_i, A, C_j, C'_j, S_j)$  to the auction server.
6. The auction server stores this information from  $EB_j$  as a tuple  $(EB_j, K_j)$  in its bid table.

### 2.3.3 Pre-filtering Phase

The goal of bids pre-filtering is to quickly identify potential winner or winners from all the bids in the AS's bid table.

For example, if  $DER_i$  wants to filter the bids for energy  $GID_i$  according to the user's reputation  $w'_i$ ,  $DER_i$  generates a trapdoor  $t_{w'_i}$  in advance and sends it to the RS. In order to preserve the privacy of  $DER_i$  and  $EB_j$ , the trapdoor  $t_{w'_i} = H_1(w'_i)^{1/(x_i)}$  is a ciphertext of the value  $w'_i$ . Then RS will send  $t_{w'_i}$  to the AS. On receiving the message from RS, for each bid in the AS's bid table, the auction server will test if the given  $C'_j$  satisfies the selection criterion  $t_{w'_i}$  of  $DER_i$ :

1. Message verification:
  - (a) The auction server verifies signature  $S_j$  on message  $(C_j, C'_j)$  with respect to the public key  $A$ .
  - (b) If it fails, the auction server will reject this bid; else the auction server will go on testing.
2. Trapdoor and tag test:
 

The auction server tests if  $H_3(e(B_j, t_{w'_i})) = F_j$ . If so, which means  $w_j = w'_i$ ; and the encrypted bid  $C_j$  will be stored in a filtered array  $W[]$ . Later,  $W[]$  will be transferred to the RS. If not, AS will go on testing the other bids. The correctness of  $H_3(e(B_j, t_{w'_i})) = F_j$  is as follows:

$$\begin{aligned}
H_3(e(B_j, t_{w'_i})) &= H_3(e((g^{x_i})^{r_j}, H_1(w'_i)^{1/(x_i)})) \\
&= H_3(e(g, H_1(w'_i)^{r_j})) \\
&= F_j = H_3(e(g, H_1(w_j)^{r_j}))
\end{aligned}$$

### 2.3.4 Decision-of-Winner Phase

On receiving filtered bids array  $\mathbf{W}[]$  from the AS, the RS can decrypt each  $C_j$  in  $\mathbf{W}[]$  as  $bid_j = C_j \oplus H_3(e(B_j, H_2(A))^{1/x_i})$  by using  $DER'_i$  secret key  $x_i$ , otherwise,  $C_j$  will be discarded. The correctness of the decryption is shown as follows,

$$\begin{aligned}
C_j \oplus H_3(e(B_j, H_2(A))^{1/x_i}) &= H_3(e(g, H_2(A)^{r_j})) \oplus bid_j \oplus H_3(e((g^{x_i})^{r_j}, H_2(A))^{1/x_i}) \\
&= H_3(e(g, H_2(A)^{r_j})) \oplus bid_j \oplus H_3(e(g, H_2(A)^{r_j})) = bid_j
\end{aligned}$$

We assume that there are  $t$  decrypted bids, and the bids will be put in a sorted array list  $\mathbf{B}[]$  according to their price descending order. Due to the special difficulties in energy storage and profit maximization of the auction in nature, the winner-selection criterion from  $DER_i$  should achieve two goals: one is that the total sales should be as high as possible; the other is that the sum of the demanded amount of the winners should be as close to the available energy demand  $Am_i$  as possible. The selected winners will be stored in an array list  $\mathbf{S}[]$  by using Algorithm 1. Finally, the RS will secretly deliver the winners list  $\mathbf{S}[]$  to  $DER_i$ .

## 2.4 Extended SESA with Conjunctive Keywords Search

The SESA can be extended to support conjunctive keywords search, with which the DERs can get more detailed information about the bids. Since the Decision-of-winner phase in this extension is same as that in SESA, we only introduce the Registration phase, Bidding phase and Pre-filtering phase as following.

### 2.4.1 Registration Phase

In this extended scheme there are also  $n$  DERs and  $m$  EBs in the energy marketing. For each  $DER_i$  ( $i = 1, \dots, n$ ) and  $EB_j$  ( $j = 1, \dots, m$ ), when they register, the RS picks two random numbers  $x_i, y_i \in \mathbb{Z}_q^*$  and sets  $pd_i = g^{x_i}$ ,  $pb_j = g^{y_j}$ .  $(pd_i, x_i)$  and  $(pb_j, y_j)$  are  $DER_i$ 's and  $EB_j$ 's public/private key pairs respectively.

**Algorithm 1** Winner selection(**B**,**S**)

---

```

1:  $c \leftarrow Am_i$ ,  $c$  is the remain energy amount;
2:  $k1, k2 \leftarrow 0$ ;  $k = t$ ;  $S[] \leftarrow \phi$ .
3: If two bids have the same price, the one requires bigger amount will be first served.
4: while  $k \neq 0$  do
5:   for each  $B[k1]$  do
6:     if ( $B[k1].price = B[k1 + 1].price$ ) and ( $B[k1].amount < B[k1 + 1].amount$ ) then
7:        $temp \leftarrow B[k1]$ ;
8:        $B[k1] \leftarrow B[k1 + 1]$ ;
9:        $B[k1 + 1] \leftarrow temp$ ;
10:    end if
11:     $k1++$ ;
12:  end for
13:   $k--$ ;
14: end while
15:  $k1 \leftarrow 0$ ;
16: for each  $B[k1]$  do
17:   if ( $B[k1].amount < c$ ) then
18:      $S[k2] \leftarrow B[k1]$ ,  $k2++$ ;
19:      $c \leftarrow c - B[k1].amount$ ;
20:   end if
21:    $k1++$ ;
22: end for
23: return ( $S[]$ );

```

---

The RS randomly chooses a master key  $s \in Z_q^*$  and assigns an ID-based key pair  $(H_1(ID_{EB_j}), H_1^s(ID_{EB_j}))$  for each  $EB_j$  ( $j = 1, \dots, m$ ). The key pair is represented as  $(vk_j, ssk_j)$ . Similar to the SESA, the  $DER_i$  will publish its energy information  $m_i = (p_i, GID_i, Ts, Lo_i, Am_i, T_N)$  publicly. The RS will store the information from each  $DER_i$  as a tuple  $(DER_i, m_i)$  in its database. Also,  $EB_j$  will register its personal information  $e_j = (Lo_j, Rep_j, Ty_j, \Delta)$  on the RS. The RS also stores the information from each  $EB_j$  as a tuple  $(EB_j, e_i)$  in its database.

In order to provide more convenience for the DERs to get detailed filtering, i.e. let them achieve the conjunctive keywords search from the auction server, each  $EB_j$  will select a keywords set  $W_j = \{w_{j1}, w_{j2}, \dots, w_{jL}\}$  to characterize his bid. Without loss of generality, the location of each type of keyword in the keywords set  $W_j = \{w_{j1}, w_{j2}, \dots, w_{jL}\}$  is fixed. For instance,  $w_1$  denotes the type of the source address keyword,  $w_2$  denotes the type of energy amount keyword etc. Keywords in the  $DER'_i$  tag and  $EB'_j$  trapdoor are in the same order.

### 2.4.2 Information Encryption

Each  $EB_j$  publishes its bid as following steps:

1.  $EB_j$  gets an identity based signature key pair as  $(vk_j, ssk_j)$ . The public key is denoted as  $A = vk_j$ , and the private key  $ssk_j$  is kept secretly.



2.  $EB_j$  selects a random number  $r_j \in Z_q^*$ , and generates a  $bid_j = (EB_j, pr_j, GID_i, Cb_j, Ts_j, \Delta)$ , where  $pr_j$  is the price of the bid,  $Cb_j$  is the amount of the energy that  $EB_j$  want to buy,  $\Delta$  is the other information of  $EB_j$ . Then  $EB_j$  computes  $C_j = H_3(e(g, H_2(A)^{r_j})) \oplus bid_j$ .
3.  $EB_j$  computes a tag for each keyword as  $t_{jk} = e(g, H_1(w_{jk})^{r_j})$ , ( $k = 1, \dots, L$ ),  $B_j = (g^{x_i})^{r_j}$ .  $EB_j$  outputs  $C'_j = (B_j, t_{jk} (k = 1, \dots, L))$ .
4.  $EB_j$  generates a signature  $S_j = S_{ssk}(C_j, C'_j)$ , where the message to be signed is the tuple  $(C_j, C'_j)$ .
5.  $EB_j$  sends the encrypted messages  $K_j = (A, C_j, S_j, C'_j)$  to the auction server.
6. The auction server will store this information from  $EB_j$  as a tuple  $(EB_j, K_j)$  in its bid table.

### 2.4.3 Pre-filtering Phase

If the  $DER_i$  needs to filter the bids by using some criteria (e.g. reputation, location etc.). It will generate a keywords set  $Q_i = \{w_{E1}, w_{E2}, \dots, w_{Et}\}$ . Then  $DER_i$  generates a trapdoor  $t_{Q_i}$  and sends it to the RS. At the end of the auction, the RS will transfer this trapdoor  $t_{Q_i}$  to the AS to filter the bids. Without loss of generality, we assume,  $\{E1, E2, \dots, Et\}$  is the subset of  $\{j1, j2, \dots, jL\}$ .

1.  $DER_i$  generates a trapdoor on the keywords  $Q_i$  as  $t_{Q_i} = (H_1(w_{E1}).H_1(w_{E2}) \dots H_1(w_{Et}))^{1/(x_i)}$ .  $DER_i$  sends  $(t_{Q_i}, \{E1, E2, \dots, Et\})$  to the RS. The RS transfers them to the AS.
2. For each  $C_j$  in  $GID_i$ 's bid table, the auction server will test if  $C'_j$  satisfies the  $EB_j$ 's requirement:

(1) Message verification:

- (a) The auction server verifies signature  $S_j$  on message  $(C_j, C'_j)$  with respect to the public key  $A$ .
- (b) If it fails, the auction server will reject this bid; else the auction server will go on testing.

- (2) The AS tests if  $H_3(e(B_j, t_{Q_i})) = H_3(\prod_{v=E1}^{Et} t_v)$ . If so,  $C_j$  will be stored in an array list  $W[]$ ; if not,  $C_j$  will be rejected. The correctness of the test is shown as follows:

$$\begin{aligned}
 H_3(e(B_j, t_{Q_i})) &= H_3(e((g^{x_i})^{r_j}, (H_1(w_{E1}).H_1(w_{E2}) \dots H_1(w_{Ek}))^{1/(x_i)})) \\
 &= H_3(e(g, H_1(w_{E1})^{r_j}).e(g, H_1(w_{E2})^{r_j}) \dots e(g, H_1(w_{Ek})^{r_j})) = H_3(\prod_{v=E1}^{Ek} t_v)
 \end{aligned}$$

## 2.5 Security Analysis

In this subsection, we analyze the security properties of the proposed SESA scheme. In particular, following the security requirements discussed earlier, our analysis will focus on how the proposed SESA scheme can achieve the goals. The extension can also achieve these properties.

- *The individual EB's bid is privacy-preserving in the proposed SESA:* In the proposed SESA scheme, EB's bidding information is encrypted by its secret number  $r_j$  as  $C_j = H_3(e(g, H_2(A)^{r_j})) \oplus bid_j$ . Anyone, including the auction server who does not know the secret number  $r_j$  can not recover  $bid_j$  from the ciphertext  $C_j$ . Thus, if a bidder does not win the auction, in the proposed SESA nobody can get any information about the bidder from its bid.
- *The authentication and data integrity of the individual EB's bid is achieved in the proposed SESA:* In SESA, each EB's bidding information is signed by the identity based signature scheme [15]. Since the identity based signature  $S_j = S_{ssk}(C_j, C'_j)$  is provably secure, the source authentication and data integrity can be guaranteed. As a result, the adversary  $\mathcal{A}$ 's malicious behaviors in the smart grid communications can be detected in the proposed SESA.
- *The EB's keyword privacy and DER's trapdoor privacy are also achieved in the proposed SESA:* In the proposed SESA, on one hand, the keyword which EB chose to append on the encrypted bid is protected by a hash function. Anyone, including the AS, can not recover  $w_j$  with the message  $C'_j$ . On the other hand, when RS delivers DER's query to the AS to search for certain type of bids, the query is also not delivered by plaintext, it is protected by a hash function. Thus, anyone gets the trapdoor only know the hash value of the keyword  $w'_i$ , and they do not know what the DER is really inquiring. Even when the AS does the verification of the tag and the trapdoor, it can not know anything about the keyword except for whether the they match or not.
- *The DER's trapdoor can not be forged in the proposed SESA:* In the proposed SESA, although the AS can get lots of trapdoors from DERs, it can not forge a valid new one from the existing old ones. That is because all the keywords are blinded by a hash function, the AS can not get the real value of the keywords.

**Table 2.1** Comparison of security properties

Properties	Scheme [6, 16]	Scheme [12]	Scheme [13]	SESA [17]
Confidentiality	No	Yes	Yes	Yes
Data privacy	No	No	Yes	Yes
Bid integrity	No	Yes	Yes	Yes
Keyword privacy	No	No	No	Yes
Trapdoor unforgeability	No	No	No	Yes

It is illustrated in Table 2.1 that most of the auction schemes [6, 16] for power market are lack of security concerns. While in traditional electronic auction system, the work in [12] only achieves the confidentiality and data integrity, the work in

[13] achieves confidentiality, data privacy and data integrity. Only the proposed SESA scheme can achieve additional keyword privacy and trapdoor unforgeability compared with [13].

Figure 2.2 shows that if the auction server is compromised, the bids information and bidder's privacy will be disclosed in schemes [6, 13, 16], only those in [13] and the proposed SESA scheme can remain secure. But [13] can not support keyword search on the bids, and there is only one winner in [13]; it is not applicable for energy auction in the smart grid. From the above analysis, we can see the proposed SESA scheme can provide enough security guarantees for auction in smart grid marketing.

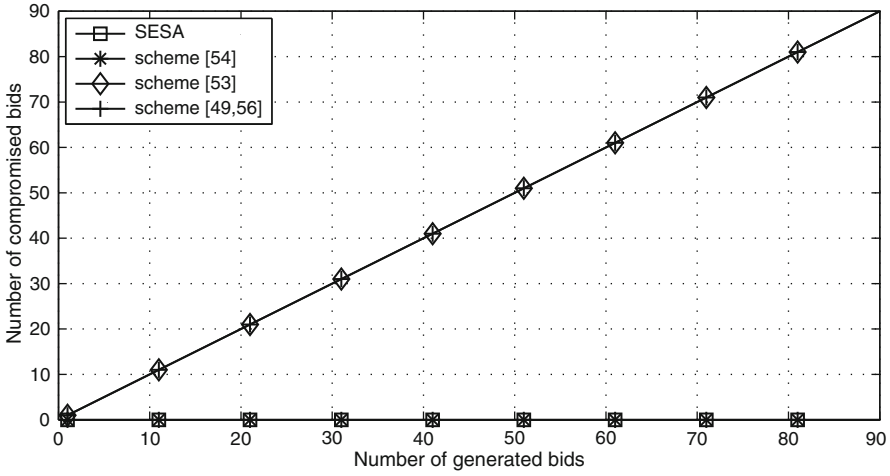


Fig. 2.2 Fraction of compromised bids when the auction server is compromised

## 2.6 Performance Analysis

### 2.6.1 SESA vs. EPPKS

In this subsection, we will compare our SESA with the privacy preserving keyword search scheme (EPPKS) [18] in terms of the computation and communication overhead in the one keyword search process.

*Computation:* In our proposed SESA, the computation tasks include pairing operations and exponentiation operations, where the pairing operations are the most time-consuming tasks. Since the hash operation and number multiplication are too fast compared with the pairing operations, we will not take them into consideration in this subsection. For simplicity of description, the pairing operation and exponentiation operation are denoted as  $C_p$  and  $C_e$ , respectively.

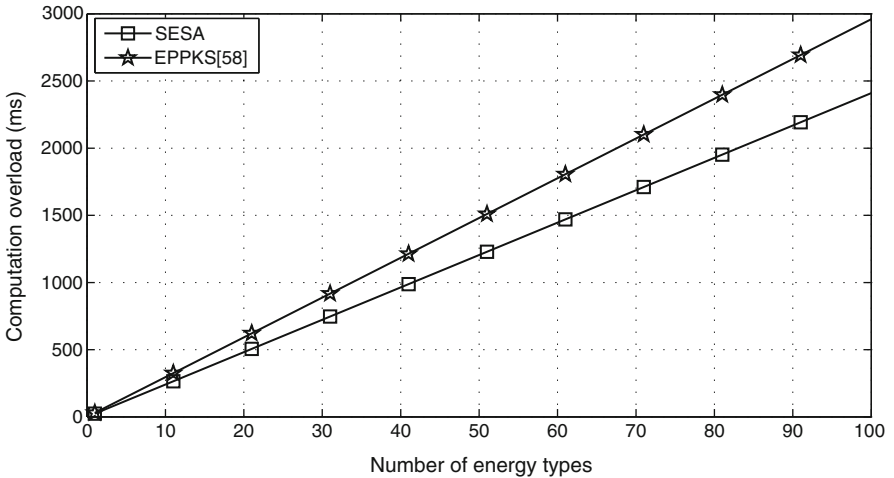
For the proposed SESA scheme, when an energy buyer  $EB_j$  generates an encrypted bid  $(A, C_j, S_j, C'_j)$ , it requires 3 exponentiation operations and 2 pairing operations for bid encryption generation, i.e.  $2C_p + 3C_e$ . The  $DER_i$  or the RS needs 1 exponentiation operations to compute a trapdoor  $t_{w'_i}$ . After receiving the trapdoor from  $DER_i$ , the local AS needs to compute 2 pairings to verify the signature [15] and 1 pairing to test if there is a bid satisfying  $DER_i$ 's query. Finally,  $DER_i$  or the RS requires 1 pairing operation and 1 exponentiation operation to decrypt the ciphertext if there are suitable bids.

In comparison, for EPPKS [18], it needs 3 pairing operations and 6 exponentiation operations to generate a data encryption on one keyword, i.e.  $3C_p + 6C_e$ . The seeker needs 1 exponentiation operation to compute a trapdoor  $T_{w_i}$ . And the server needs 1 pairing operation to test whether a given tag contains keyword  $T_{w_i}$ . Then the server needs  $2C_p + 2C_e$  more computation overhead to get an intermediate result of the partial decipherment. At last, it will cost the seeker  $C_e$  to recovery the ciphertext.

**Table 2.2** Comparison of computation complexity

	SESA	EPPKS
EB	$2C_p + 3C_e$	$3C_p + 6C_e$
AS	$3C_p$	$3C_p + 2C_e$
$DER_i$ or RS	$C_p + 2C_e$	$2C_e$

Table 2.2 indicates that SESA is more efficient than EPPKS [18]. Detailed experiments also are conducted on a Pentium IV 3 GHz system to study the execution time [19]. For  $G_1$  over the FST curve, a single exponentiation operation in  $G_1$  with 161 bits costs 1.1 ms and the corresponding pairing operation costs 3.1 ms. The comparison of computation overhead is shown in Fig. 2.3. We can see that SESA achieves totally lower execution times compared to EPPKS. Moreover, SESA can guarantee the integrity of the message, while EPPKS can not achieve this property.



**Fig. 2.3** Comparison of computation overhead between SESA and EPPKS

*Communication:* Most pairing-based cryptosystems need to work in a subgroup of the elliptic curve  $E(F_q)$ . By representing elliptic curve points using point compression [20], the length of the elements in  $G_1$  and  $G_2$  will be roughly 161-bit (using point compression) and 1,024-bit, respectively. SHA-1 is used to compute the hash function, which yields a 160-bit output. Let the parameter  $n$  in EPPKS be 160-bit. The communications among the three entities of the proposed SESA can be divided into three parts, EB-to-AS, DER-to-AS, and AS-to-RS communications.

We first consider the EB-to-AS communication in SESA. In the information encryption phase, the data report is in the form of  $K_j = (A, C_j, S_j, C'_j)$ . Since the length of identity based signature [15] is two  $G_1$  elements, the size of  $K_j$  should be  $160 + 160 + 161 * 2 + 160 + 161 = 963$  bits. In the DER-to-AS communication, DER needs to delivery a trapdoor  $t'_w$  to the AS, which is 160 bits; while in AS-to-RS communication, the AS will reply a ciphertext  $C_j$  to the EB if there is energy matching EB's demand, which is 160 bits.

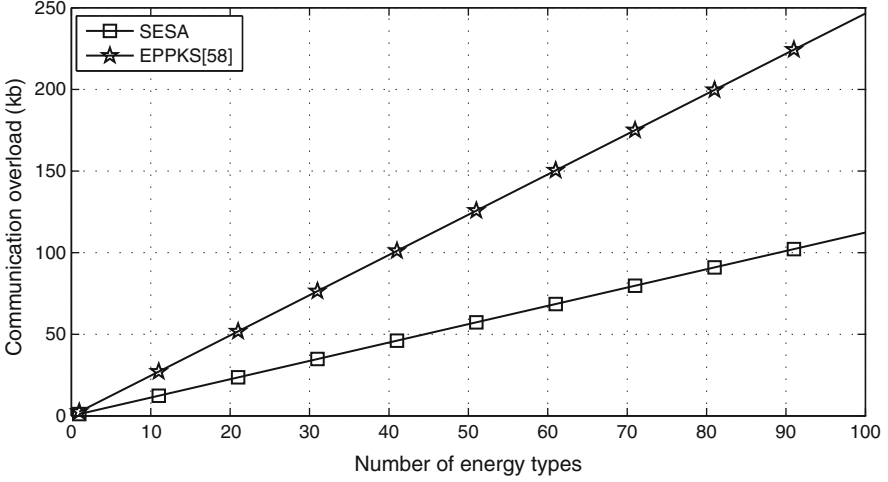
**Table 2.3** Comparison of communication complexity(bits)

	SESA	EPPKS
EB-to-AS	803	640
DER-to-AS	160	160
AS-to-RS	160	1,665

In contrast, the user-to-server communication overhead in EPPKS is the message  $(C_m, C_w)$ , which includes one  $G_1$  element, two  $n$ -bit elements and one hash element. The size is  $161 + 2n + 160 = 641$  bits. Then, the trapdoor  $T_{w_j}$  with the size of 160-bit will be sent from user to the server. In the server-to-receiver communication, if there is a keyword match, the server will reply  $(C_m, C_\rho, C_w)$  to the receiver. Here,  $C_\rho$  is an element of  $G_2$ . The size of the reply is  $161 + 160 + 2n + 1,024 = 1,665$  bits. Table 2.3 and Fig. 2.4 show the comparison of communication overhead between SESA and EPPKS. It can be seen that the SESA scheme significantly reduces the communication overhead.

### 2.6.2 Extended SESA vs. EPPKS

In this subsection, we will compare our extension of SESA with the privacy preserving keyword search scheme (EPPKS) [18] in terms of the computation overhead in the conjunctive keywords search process. Suppose there are 10 keywords tags on each bid, and 5 keywords in the  $EB'_j$  conjunctive search trapdoor. In the extension, it costs the  $EB_j$   $10 + 1$  pairing operations and  $10 + 2$  exponentiation operations to generate an energy encryption  $(A, C_j, S_j, C'_j)$ . That is  $11C_p + 13C_e$ . while the  $DER_i$  or the RS needs  $5 + 2$  hash operations and 1 exponentiation operation to compute the trapdoor. On receiving the trapdoor  $t'_w$  from RS, the local AS needs 5 pairing operations and 1 hash operation to test  $DER_i$ 's query. If there is a suitable bid, the local AS needs to compute 2 pairings to verify the signature



**Fig. 2.4** Comparison of communication overhead between SESA and EPPKS

and 1 pairing to test if there is an energy satisfy  $DER_i$ 's load demand. The  $DER_i$  or RS requires 1 pairing operation and 1 exponentiation operation to decrypt the ciphertext.

**Table 2.4** Comparison of computation complexity

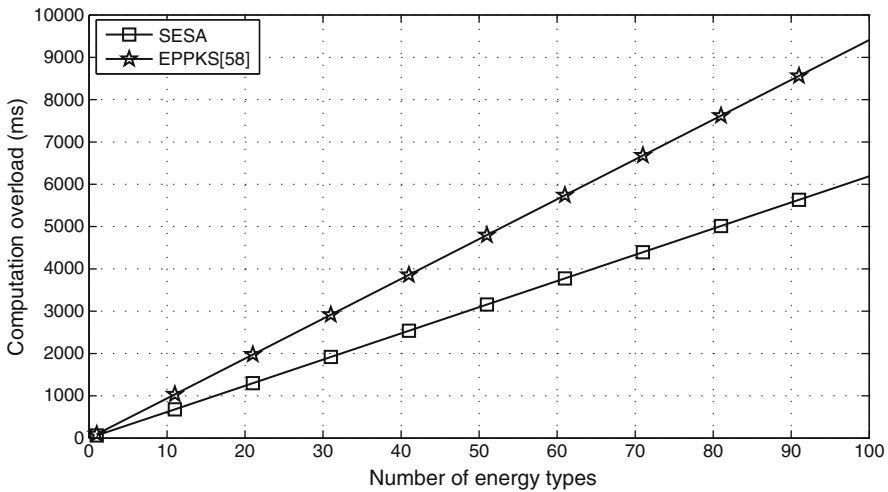
	SESA	EPPKS
EB	$11C_p + 12C_e$	$12C_p + 24C_e$
AS	$3C_p$	$7C_p + 2C_e$
$DER_i$ or RS	$C_p + 2C_e$	$6C_e$

In comparison, the EPPKS needs  $10 + 2$  pairing operations and  $10 * 2 + 4$  exponentiation operations to generate an energy encryption on 10 keywords. That is totally  $12C_p + 24C_e$ . Since EPPKS can do 1 keyword search at a time, for 5-keyword search, the seeker needs to compute 5 trapdoors and sends them to the server, which needs 5 exponentiation operations. Thus, the server needs to test 5 times. Each time, the server needs 1 pairing operation to test whether a given tag contains keyword  $T_{w_i}$  or not. Thus, the server totally needs  $5C_p$  to test all of the trapdoors. If there is a matching item, the server needs  $2C_p + 2C_e$  more computation overhead to get an intermediate result of the partial decipherment. At last, it will cost the seeker  $C_e$  to recovery the ciphertext.

From Table 2.4 and Fig. 2.5, it can be seen that the extension of SESA requires much less computation overhead than the EPPKS for the conjunctive keywords search. In addition, the extension is also more efficient than the EPPKS in terms of communication overhead, because more trapdoors need to be sent to the server in EPPKS.

## 2.7 Related Works

The traditional auction schemes can be divided into two categories: open outcry and sealed bids. Open outcry can further be separated into English auctions and Dutch auctions [16]. In English auctions, the value of the bid is public, and the price of the bid must be higher than the current price. The highest bidder is the winner at the end of the bidding phase. There are many famous English auction web sites (e.g., Yahoo!, eBay, etc.) [13]. The Dutch auction is almost the same as the English auction, except that it begins with the top price. In a sealed bid auction, the bidders write the price and quantity of their bid on a sheet of paper, and then they seal the sheet and give it to the auctioneer. The auctioneer collects all the sealed sheets and opens them after the deadline to determine the winner. A sealed bid auction can be separated into two kinds, first-price sealed-bid and second-price sealed-bid.



**Fig. 2.5** Comparison of computation between extended SESA and EPPKS

The bidding manner has been extensively studied and various bidding models are presented in the power market [5,6]. Among the various methods, the simplest way is to estimate the market clearing price of the next time and then present the bid with a lower price than the estimated one. The second method is to estimate the behaviors of the rivals and to present the bid [6]. The third method is based on the game theory [21] with oligopolistic strategy such as Cournot model, and supply function models [5]. But, few of them considers the privacy of the bidders and the energy providers. In electronic auction systems, Chang [12] and Li [13] both presented anonymous auction protocol with freewheeling bids. However, bidding privacy can not be achieved in [12], and both of them can not support keyword search or any other filtering.

The concept of public key encryption with keyword search (PEKS) was proposed by Boneh et al. [22], which supports the keyword search on encrypted data. Other schemes focusing on constructing keyword encryption were extensively discussed, such as [23]. PECSK [24] supports conjunctive-subset keywords search. But it is only a keyword search scheme. EPPKS [18] presented a privacy preserving keyword search scheme in cloud computing. It is one of the few schemes which integrates both the message encryption and keyword search properties. However, when the server finds a tag matching the trapdoor in EPPKS [18], the server has to compute an intermediate result to help the user to recover the message, which costs communication and computation overhead.

## 2.8 Summary

In this chapter, we have studied the security and privacy concerns associated with energy auction in smart grid marketing, and proposed an efficient Searchable Encryption Scheme for Auction. We use public key encryption with keyword search to enable the energy sellers to inquire potential winner from the auction server while preserving the privacy of the EBs. In addition, an extension of SESA was presented to support detailed filtering of the bids. Security and performance analysis demonstrate that our proposed SESA and its extension both can achieve data and keyword privacy, bid integrity and trapdoor unforgeability, and they are more efficient than the existing keyword search approach EPPKS in terms of computation and communication overhead. However, for the multidimensional data in smart grid, in some cases, the conjunctive keyword query needs to support subset keywords query function for flexible usage. In the subsequent chapters, we will consider this problem and address the conjunctive query over encrypted multidimensional data.

## References

1. C. Yuen, A. Oudalov, and A. Timbus, "The provision of frequency control reserves from multiple microgrids," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 173–183, 2011.
2. B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4603–4612, 2011.
3. V. Forte, "Smart grid at national grid," in *Proc. ISGT*, pp. 1–4, IEEE, 2010.
4. S. Chakraborty, M. D. Weiss, and M. G. Simões, "Distributed intelligent energy management system for a single-phase high-frequency ac microgrid," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 97–109, 2007.
5. E. Bompard, W. Lu, and R. Napoli, "Network constraint impacts on the competitive electricity markets under supply-side strategic bidding," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 160–170, 2006.



6. Y.-q. SONG, L.-w. JIAO, Y.-x. NI, F.-s. WEN, Z.-j. HOU, and F.-l. WU, "An improovement of generation firms' bidding strategies based on conjectural variation regulation via dynamic learning," *Proceedings of the Csee*, vol. 12, p. 004, 2003.
7. X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 38–45, 2012.
8. Z. M. Fadlullah, N. Kato, R. Lu, X. Shen, and Y. Nozaki, "Toward secure targeted broadcast in smart grid," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 150–156, 2012.
9. M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.
10. R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.
11. H. Li, X. Liang, R. Lu, X. Lin, and X. Shen, "Edr: an efficient demand response scheme for achieving forward secrecy in smart grid," in *Proc. GLOBECOM*, pp. 929–934, IEEE, 2012.
12. Y.-F. Chang and C.-C. Chang, "Enhanced anonymous auction protocols with freewheeling bids," in *Proc. AINA*, vol. 1, pp. 6–11, IEEE, 2006.
13. M.-J. Li, J. S.-T. Juan, and J. H.-C. Tsai, "Practical electronic auction scheme with strong anonymity and bidding privacy," *Information Sciences*, vol. 181, no. 12, pp. 2576–2586, 2011.
14. J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, no. 13, pp. 2576–2587, 2010.
15. B. Libert and J.-J. Quisquater, "The exact security of an identity based signature and its applications.," *IACR Cryptology ePrint Archive*, vol. 2004, p. 102, 2004.
16. H.-T. Liaw, W.-S. Juang, and C.-K. Lin, "An electronic online bidding auction protocol with both security and efficiency," *Applied mathematics and computation*, vol. 174, no. 2, pp. 1487–1497, 2006.
17. M. Wen, R. Lu, J. Lei, H. Li, X. Liang, and X. S. Shen, "Sesa: an efficient searchable encryption scheme for auction in emerging smart grid marketing," *Security and Communication Networks*, vol. 7, no. 1, p. 234–244, 2013.
18. Q. Liu, G. Wang, and J. Wu, "An efficient privacy preserving keyword search scheme in cloud computing," in *Proc. CSE*, vol. 2, pp. 715–720, IEEE, 2009.
19. M. Scott, "Efficient implementation of cryptographic pairings," in [Online]. <http://www.pairing-conference.org/2007/invited/Scottslide.pdf>, 2007.
20. S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.
21. D.-J. Kang, B. H. Kim, and D. Hur, "Supplier bidding strategy based on non-cooperative game theory concepts in single auction power pools," *Electric power systems research*, vol. 77, no. 5, pp. 630–636, 2007.
22. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Eurocrypt*, pp. 506–522, Springer, 2004.
23. X. Lin, R. Lu, K. Foxton, and X. S. Shen, "An efficient searchable encryption scheme and its application in network forensics," in *Proc. E-Forensics*, pp. 66–78, Springer, 2011.
24. B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 262–267, 2011.

Querying over Encrypted Data in Smart Grids

Wen, M.; Lu, R.; Liang, X.; Lei, J.; Shen, X.S.

2014, IX, 78 p. 22 illus., 17 illus. in color., Softcover

ISBN: 978-3-319-06354-6