




An Efficient and Secure Range Query Scheme for Encrypted Data in Smart Grid

Xiaoli Zeng¹, Min Hu¹, Nuo Yu^{1,2}(✉) , and Xiaohua Jia¹

¹ Harbin Institute of Technology Shenzhen Graduate School,
Shenzhen 518055, China
yunuohit@gmail.com

² School of Electrical Engineering, Anhui Polytechnic University,
Wuhu 241000, China

Abstract. In smart grid information systems, the electricity usage data should be audited by data users, such as the market analysts to finish their tasks. Besides that, electricity company always outsources the data to the cloud server (CS) to release its data management pressure. Since the CS is untrusted and the detailed electricity usage data contains users' privacy, the privacy concern of the data and data users' queries is raised. Although many schemes have been proposed to achieve the encrypted data query in smart grid, they are not applied well due to the numeric attributes in electricity usage data and privacy concern in smart grid application. In this paper, we provide an efficient privacy-preserving scheme for range query in smart grid. Our scheme achieves the range query without disclosing the privacy of the data and queries. And the performance shows that our scheme can reduce the computation cost for both the data owner and data users, and shorten the response time of every query, which is great significance for smart grid application.

Keywords: Smart grid · Privacy-preserving · Range query

1 Introduction

With the rapid development of industrial and economic activities, smart grid has been accepted by more and more people due to its many good features. However, the electricity usage data of customers in smart grid is surging from 10,780 terabytes (TB) in 2010 to over 75,200 TB in 2015 [1]. That is far beyond the electricity company's data management capability. Uploading the electricity usage data into a cloud server is the best way to mitigate this stress. In this approach, electricity company can store the electricity usage data on cloud server and execute computation and queries using the server's computational capabilities.

However, cloud server is often untrusted. It may share the electricity usage data with other parties for profit making. But the electricity usage data contains

user's private information, e.g., user's name and family address, bank account and telephone number. If the cloud server shares these data with attackers, user's privacy might be compromised. Therefore, our electricity usage data must be stored in encrypted form on the cloud server to protect the data confidentiality and privacy.

In addition, electricity usage data in smart grid information systems should be periodically audited to ensure that the billing and pricing statements are presented fairly [2]. Specially, data users, such as market analysts, are endowed with the task of querying smart grid information systems for auditing, analysis, accounting or tax-related activities [3]. Thus, there is growing need to achieve querying on encrypted data in smart grid.

It is not a trivial issue to query on encrypted data in smart grid at the same time with the following requirements: (1) Confidentiality and privacy of data. The electricity usage data should be protected from being stolen by the untrusted cloud server. (2) Privacy of the query. Since the cloud server is untrusted, it might trace the query results if the query contains sensitive information and make the user's privacy disclosure. Thus, guaranteeing query privacy is also important for smart grid application. (3) Achieving range query. Since the electricity usage data always has the numeric attributes, range query is a common type of queries for the smart grid. (4) Being efficient and low cost. Smart grid is a large-scale system, since the electricity usage data is large and dynamic update in the cloud server, the protocol should be efficient for the query and low cost for both the data owner and data users.

Recently, many protocols were proposed to achieve the query on encrypted data, but they are not suitable to apply for the smart grid. Public key encryption with keyword search (PEKS) is a widely studied approach to achieve querying on encrypted data. Nevertheless, most of the existing schemes (such as [4, 5]) about PEKS focus only on the keyword search technique, with little attention to both data and query privacy protection in the scheme. Baek et al. [6] argue that PEKS and data encryption schemes need to be treated as a single scheme to securely provide PEKS service. Qin et al. [7] propose an efficient encryption scheme with one-dimension keyword search (EPPKS) for cloud computing by combining the ideas of partial decipherment with the PEKS. However, it is not quite secure because the partial decipherment will leak partial information of users' data. The Searchable Encryption Scheme for Auction (SESA) [8] in smart grid achieved the security, but it only can be applied for the equality checks.

In this paper, we propose a privacy-preserving range query scheme over encrypted electricity usage data for smart grid, which ensures to secure the data confidentiality, privacy and query privacy in smart grid. We first proposed a range query scheme in smart grid by using the modified Paillier homomorphic cryptosystem. With our scheme, the range query is achieved without disclosing the privacy of the electricity usage data and query context. We then evaluated the performance of our scheme. The results show that our scheme can reduce the computation cost for both the electricity company and data users, and shorten the response time of every range query, which is great significance for smart grid application.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes the system model, data query model, security requirements and our design goals. Section 4 introduces the background. Sections 5 and 6 present the modified paillier homomorphic cryptosystem and our scheme respectively. Section 7 discusses how the proposed scheme meets our design goals, and Sect. 8 shows the experiment results. Finally, concluding the paper in Sect. 9.

2 Related Work

Querying encrypted data in smart grid is an important issue that attracts great attention from research communities. But the most existing schemes only can be applied for equality checks. Since the encrypted electricity usage data has many numeric attributes, it is much significant to achieve range query in smart grid.

For the encrypted data query, there are generally four categories of solutions that have been developed for range query: (1) Order preserving encryption (OPE)-based schemes; (2) Predicate encryption-based schemes; (3) Asymmetric scalar-product preserving encryption (ASPE)-based schemes; (4) Bucketization-based schemes.

Order preserving encryption (OPE)-based schemes [9–11] that preserve the relative ordering of data items even after encryption. Agrawal et al. [9] describe the first order preserving encryption scheme for numeric data, followed by [10] which gives a formal security analysis and proposes the Order Preserving Symmetric Encryption (OSPE). Boldyreva et al. [11] revise and improve the security of OPE. The OPE scheme allows direct translation of range predicate from the original domain to the domain of the ciphertext. However, OPE encryption is deterministic and thus it reveals the frequency of each distinct value and is susceptible to statistic attacks.

In predicate encryption-based schemes [12–15] secret keys correspond to predicates and ciphertexts are associated with attributes. The secret key corresponding to a predicate can be used to decrypt a ciphertext only if the attribute satisfies the predicate. Boneh and Waters [12] propose a predicate encryption, named Hidden Vector Encryption (HVE), which can be used for range queries. To improve the search efficiency, tree-based index structures [15, 16] were proposed to support multi-dimensional range query [13]. But in those schemes, the cost to compute exponentiation and pairing in group is too high.

Asymmetric scalar-product preserving encryption (ASPE)-based schemes [17, 18] that allow the relative distance comparison between two data points under encryption. Given two data points p_1 , p_2 and a query point Q , all encrypted, ASPE can determine whether Q is closer to p_1 or p_2 . Wang et al. [17] create a hierarchical encrypted index, which first constructs a regular R-tree for a given set of data points and then applies the ASPE to encrypt the minimum bounding box range (MBR) in the R-tree. This tree-based ASPE solution reduces the leakage of sorted information, but it can cause false positives.

The bucketization technique is firstly designed in [19] for query processing in an untrusted environment. In this bucketization-based scheme [19–21], the data

owner partitions the whole attribute domain into multiple buckets of varying sizes and assigns a unique bucket tag to each bucket using a collision-free hash function. Pairs of a bucket tag and the encrypted tuples constitute the index, which is maintained on the untrusted server. When a range query is issued by the data owner, it needs to be first determined which tags of buckets intersect the query and then all the tuples indexed by these tags will be returned by the server. Although this scheme is more efficient than the three schemes mentioned before, it always contains some false positives, the data users need to filter the mismatch after decrypting all the results, which is not suitable for application of the smart grid.

Since the schemes presented above all have some shortcomings. In this paper, we aim at providing a privacy-preserving range query scheme for encrypted electricity usage data in smart grid based on the modified paillier homomorphic cryptosystem.

3 System Model

In this section we introduce the system model, data query model, security requirements and our design goals.

3.1 System Model

In the system model, our focus is on how to outsource the users' electricity usage data from the electricity company to cloud server (CS) in encrypted form and how to operate a query over the encrypted electricity usage data in CS by data users. Our system is composed of three components, as shown in Fig. 1: electricity company, data users (such as the market analysts, auditors) and a cloud server (CS).

The electricity company is the data owner, who encrypts the electricity usage data of customers by using cryptosystem before outsourcing the data to CS. And the data user always need to query the electricity usage data for their tasks. CS is honest but curious, it might be interested in users' electricity usage data and data users' queries.

3.2 Data Query Model

Before we discuss the security requirements and our design goals, let us first introduce how the encrypted data is stored at the CS and how data users make queries.

We consider relational databases, where data are represented in the form of tables. Let $R(A_1, A_2 \cdots A_n)$ be a relational table, where $A_1, A_2 \cdots A_n$ are attributes of the table. The encrypted form of the table is as following:

$$R^s(A_1^s, A_2^s \cdots A_n^s),$$

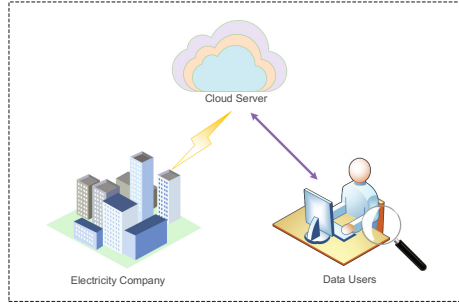


Fig. 1. System model in our scheme.

Table 1. User information table (*UIT*)

ID	Name	Address	Consumption
23	Tom	Maple	40
860	Mary	Main	80
320	John	River	50
875	Jerry	Hopewell	110

where $A_1^s, A_2^s \dots A_n^s$ are encrypted attributes. For example, consider the *UIT* table below that stores the information of customers (Table 1).

The *UIT* table is mapped to a corresponding *UIT^s* table at the CS:

$$R^s(ID^s, Name^s, Address^s, Consumption^s)$$

where $ID^s, Name^s, Address^s, Consumption^s$ denote encrypted strings of the ID, Name, Address and Consumption respectively. For instance, the following is the encrypted table *UIT^s* stored on the CS (Table 2):

Table 2. *UIT^s*

ID^s	$Name^s$	$Address^s$	$Consumption^s$
1100...	0111...	0001...	0100...
0110...	0011...	0101...	0111...
0010...	1111...	0100...	1000...
1110...	0000...	1001...	1101...

The column strings contain the vaules corresponding to the encrypted values in *UIT*. For instance, the first vaule is encrypted to “1100...” that is equal to *encrypt* (23), the second vaule is encrypted to “0111...” that is equal to *encrypt* (Tom).

In this model, data users use the SQL statements to query the encrypted data. For example, data users use:

```
SELECT Name, Address, Consumption
FROM UIT table
WHERE Consumption>100;
```

and the client software at userside will translate this SQL query Q into an encrypted form Q^s :

```
SELECT Names, Addresss, Consumptions
FROM UITs table
WHERE Consumptions>100s;
```

where $Name^s$, $Address^s$, $Consumption^s$, 100^s are the ciphertext of the respective strings. It is then submitted to CS for execution. CS will return encrypted data that satisfy the SQL conditions to the user.

The conditions of the SQL statements can be classified to two categories:

- (1) Attribute = Value. Such condition is equality query, like consumption = 80;
- (2) Attribute > Value or Attribute < Value. Such condition is range query. For instance, consumption > 70 or consumption < 60.

Since extensive research has been done on equality condition on encrypted data, we focus on range query in this paper.

3.3 Security Requirements

As mentioned before, in system model, CS might be interested in the electricity usage data. It has the motivation to steal the individual data for its own purpose. In addition, it might trace or analyze the query results, if the query contains sensitive information. Therefore, our scheme should satisfy the following security requirements.

Data Confidentiality: The electricity company should encrypt the electricity usage data before uploading it to the CS, and successfully prevents the CS from stealing the data.

Data privacy: The encrypted electricity usage data should be accessed only by authenticated data users. It means that only the authorized data users can decrypt the encrypted data.

Query privacy: Data users usually prefer to keep their queries from being exposed to others. Thus, the biggest concern is to encrypt the query to protect the query privacy. Otherwise, if the query includes some sensitive information, the CS might trace or analyze the results.

3.4 Design Goals

In this model, our design goal is to develop a privacy-preserving range query scheme over encrypted electricity usage data for smart grid application, and achieves the security and efficiency as follows.

- (1) Since the CS is untrusted and the electricity usage data contains the privacy of the user, our scheme should achieve the data confidentiality and data privacy, as well as the query privacy.
- (2) In smart grid application, the electricity usage data is large and dynamic update in the cloud. As range query are operated over encrypted electricity usage data, comparing with the existing range query schemes in smart grid, our scheme should reduce the response time of every range query and reduce the computation cost for both the data owner and data users.

4 Background

In this section, we will first introduce the Paillier Homomorphic Cryptosystem which are the based of our scheme.

The Paillier homomorphic cryptosystem is a public key cryptosystem by Paillier [22] based on the “Composite Residuosity Assumption (CRA)”. The Paillier cryptosystem is homomorphic, by using a public key, the encryption of the sum $m_1 + m_2$ of two messages m_1 and m_2 can be computed from the encryption of m_1 and m_2 . Our scheme is inspired by the Paillier cryptosystem. Hence, we give some preliminaries of the Paillier homomorphic cryptosystem, which consists of three phases as follows.

Key Generation. Set $n = pq$, where p and q are two large prime numbers. Set $\lambda = lcm(p - 1, q - 1)$, i.e., the least common multiple of $p - 1$ and $q - 1$. Define $L(\mu) = \frac{\mu+1}{n}$, and randomly choose g_p , then compute

$$\mu = (L(g_p^\lambda \pmod{n^2}))^{-1} \pmod{n}.$$

The public encryption key is a pair (n, g_p) . The private decryption key is (λ, μ) .

Encryption $E(m, r)$. Given plaintext $m \in \{0, 1, \dots, n - 1\}$, select a random $r \in \{0, 1, \dots, n - 1\}$, and encrypt the plaintext m as ciphertext c :

$$c = E(m, r) = g_p^m \cdot r^n \pmod{n^2}.$$

Decryption $D(c)$

$$D(c) = L(c^\lambda \pmod{n^2}) \cdot \mu \pmod{n} = m.$$

5 Modified Paillier Cryptosystem

In our scheme, we use the Paillier homomorphic cryptosystem so that CS can perform matching operation without decrypting the electricity usage data and query contexts. In this section, we provide the details of our modified Paillier cryptosystem.

5.1 Making μ Public

Recall that in the Paillier cryptosystem, (λ, μ) is the private key. However, μ can be made public, because it is hard to decrypt an encrypted message by only knowing μ . Hence, we can make μ public while achieving the same security guaranty as the unmodified Paillier cryptosystem.

We take advantage of this operation in order to shift the computation towards encryption and make decryption lightweight.

5.2 Shifting the Computation

With the modification above, the new public key is (n, g_p, μ) and the private key is λ . First, we modify the Paillier homomorphic cryptosystem so that anyone can decrypt using the new public key, but only those holding the private key can encrypt. This is similar to the digital signatures. And the following equations show the modification to the encryption and decryption algorithms:

Encryption:

$$\begin{aligned} E'(m, r, \lambda) &= E(m, r)^\lambda \\ &= g_p^{m\lambda} \cdot r^{n\lambda} \pmod{n^2} \\ &= c. \end{aligned}$$

Decryption:

$$D(c) = L(c \pmod{n^2}) \cdot \mu \pmod{n} = m.$$

We can realize that one can perform all the homomorphic operations on our modified Paillier cryptosystem similar to the Paillier cryptosystem.

Note that as we shift the computation towards encryption, the decryption is computationally more efficient than the Paillier decryption. And we also allow the CS to perform certain operations without knowing the private key. Such shifting improves the performance of the range query model, since the Paillier decryption become more efficient.

5.3 Secret Comparisons

With the shift of computation described above, CS can find the difference by simply decrypting each value, which does not assure the privacy of individual values. Therefore, we introduce an additional parameter to the encryption operation in order to allow CS to compute the difference without knowing individual values.

Assume that there are two values x_1 and x_2 . We perform the following operation to the encryption so that CS can find the difference $(x_1 - x_2)$ without learning either x_1 or x_2 :

$$y_1 = g^t \cdot E'(x_1, r_1) \pmod{n^2},$$

$$y_2 = g^{-t} \cdot E'(-x_2, r_2) \pmod{n^2}.$$

Note that even though μ is known, it can decrypt neither x_1 nor x_2 as they are multiplied with g^t and g^{-t} respectively. Due to the homomorphic property, we can have:

$$y_1 \cdot y_2 = E'(x_1 - x_2, r_3).$$

Anyone can compute the difference as follows using the public key of the modified Paillier cryptosystem:

$$D(y_1 \cdot y_2) = x_1 - x_2.$$

The results $D(y_1 \cdot y_2) > 0$, $D(y_1 \cdot y_2) < 0$ and $D(y_1 \cdot y_2) = 0$, indicate the cases of $x_1 > x_2$, $x_1 < x_2$ and $x_1 = x_2$, respectively.

For example, if the data user wants to query the users whose electricity consumption is greater than 100, then the x_2 is 100. The CS will return the encrypted data to the user. As we can see, with this method, CS can compare two numeric values, but is unable to know the exact values of them.

6 Privacy Preserving Range Query Scheme

There is three entities in the range query model in smart grid: electricity company, data users and a CS. For each query, the scheme works in the following steps, as shown in Fig. 2:

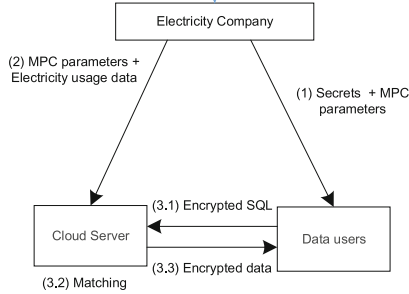


Fig. 2. The steps of range query in our system.

- (1) Initialization of the electricity company and the data user.
- (2) Electricity company uploads the encrypted electricity usage data to CS.
- (3) Data users make queries to CS and get the results.

In the proposed scheme, we aim at providing a privacy-preserving range query scheme in smart grid based on modified Paillier cryptosystem. We will explain each step in details in the following subsections.

6.1 Initialization of Electricity Company and Data Users

When the electricity company initializes, it generates the following values: $E'(r_i)$, $E'(1)$, and $g^t \cdot E'(r_i)$, which are used by the electricity company to encrypt the data before uploading them to the CS.

Besides that, during the initialization, the company checks the identify of the data user. If it is a legal user, electricity company will send the following values to it: $-r_i$, $E'(-1)$, and $g^{-t} \cdot E'(-r_i)$.

Note that these parameters are used by the data user to encrypt the queries and decrypt the results. The electricity company may provide $E'(-1)$ and $-r_i$, and allow the data user to compute $E'(-r_i)$ homomorphically, instead of providing the value directly. In this case, data user can recover neither g^{-t} nor $-t$ from $g^{-t} \cdot E'(-r_i)$.

6.2 Upload the encrypted data to CS by electricity company

When the electricity company wants to upload the data, it first encrypts the electricity usage data. We illustrate our ideas using examples. Consider the *UIT* table before, we encrypt one of the columns in the data table as an example. Let one of the consumption values as v_1 . It is encrypted to y_1 as following:

$$\begin{aligned} y_1 &= g^t \cdot E'(r_i) \cdot E'(r_i(v_1 - 1)) \\ &= g^t \cdot E'(r_i v_1). \end{aligned}$$

The encryption of other attribute values is similar to this example.

Note that $E'(r_i(v_1 - 1))$ is homomorphically computed using $E'(r_i)$. This value can be computed efficiently by using fast multiplication.

After the electricity company encrypts the electricity usage data, it uploads the encrypted data to the CS.

Note that CS cannot decrypt the encrypted data, but our scheme allows the CS to perform privacy preserving matching.

6.3 Secure Data Query by Data Users

When the data user makes a SQL query, the query is encrypted and the encrypted query is sent to the CS.

Considering the following query as an example:

```
SELECT Name, Address, Consumption
FROM UIT table
WHERE Consumption>100;
```

The value 100 is encrypted into the form 100^s in the example. We use x_1 to express the value 100 and w_1 expresses the encrypted form 100^s . The operation is as follows:

$$\begin{aligned} w_1 &= g^{-t} \cdot E'(-r_i) \cdot E'(r_i(1 - x_1)) \\ &= g^{-t} \cdot E'(-r_i x_1), \end{aligned}$$

When the CS receives the encrypted SQL query:

```

SELECT  $Name^s, Address^s, Consumption^s$ 
FROM  $UIT^s$  table
WHERE  $Consumption^s > 100^s$ ;

```

It searches data table UIT (encrypted) and compares each attribute values (encrypted) in consumption column with 100^s . It computes the difference d between each consumption value in the table with 100^s as follows:

$$\begin{aligned}
 d &= D'(y_1 \cdot w_1) \\
 &= r_i(v_1 - x_1).
 \end{aligned}$$

Since the r_i is greater than 0, CS will return the encrypted data to the data user, which makes the $d > 0$.

Note that, the electricity usage data always contains more than one attribute. If the data user queries the data more than one attribute, CS has to match for a composite range query after evaluating each rang query value.

And after successfully receiving the result, the valid data user can decrypt the encrypted data using the secrets.

7 Security Analysis

In this section, we will explain how our scheme achieves the goals of the data confidentiality, data privacy and query privacy.

7.1 Data Confidentiality

The data confidentiality in our scheme requires that the electricity usage data should be encrypted when it is uploaded to the CS, and prevents the CS from stealing. In our scheme, the electricity usage data is encrypted by Paillier cryptosystem. And as for CS, since it only does homomorphic computing on two encrypted values, it cannot access the electricity usage data. Therefore, the proposed scheme can achieve the data confidentiality.

7.2 Data Privacy

Data privacy in our scheme means that only the authorized data user can decrypt the electricity usage data. Data in our proposed scheme are encrypted by Paillier cryptosystem, so the adversary cannot identify them. But if the adversary fabricates a message and sends it to some entities, it cannot be detected. Hence, we also use the protocol in our scheme, only the data user who is authenticated by the electricity company can get the secrets to decrypt. Therefore, our proposed scheme can achieve the data privacy.

7.3 Query Privacy

The query privacy in our scheme means that the query should be encrypted to keep from being exposed to the CS. In our scheme, queries are also encrypted by the Paillier cryptosystem. When CS wants to do the matching for the electricity usage data, it does not need to know the exact value of the query. It only does homomorphic computing on two encrypted values. Thus, our proposed scheme satisfies the goal of query privacy.

8 Experiment Result

In this section, we evaluate the performance of the proposed scheme in terms of response time of a range query and the computation cost of the data owner and data users.

8.1 Response Time

In smart grid, it is important for data users to know the response time of a range query, which can benefit for them to efficiently schedule their tasks. We analyze the response time of our scheme and compare our scheme with the Bucketization-based scheme.

We implement the proposed scheme and the Bucketization-based scheme respectively in JRE 1.7, eclipse and run it in the computer in Windows 7 OS with the CPU i5 and 4 cores. We test the response time of a range query by those two schemes respectively.

From the Fig. 3, we can see that: when the data records increase in database, the response time of a range query in our scheme changes little. But the change in the Bucketization-based scheme is obvious. We can see from the Fig. 4, which is more precise: when the data records increase, the response time of a range query in Bucketization-based scheme increases nonlinearly but fast. This is a huge pressure for the data user, because the data uses have a lot of data to be audited in reality.

Therefore, we can conclude that our scheme is efficient enough to meet the requirement of smart grid application. Even the data records are large in database, the response time of our scheme will be small, which is significant for smart grid application.

8.2 Computation Cost

For the computation cost, we give the comparison between our scheme and Bucketization-based scheme too. The experimental environment is the same as the previous subsection and we choose 5000 data records. The computation cost of the data owner and data users will be introduced respectively in following.

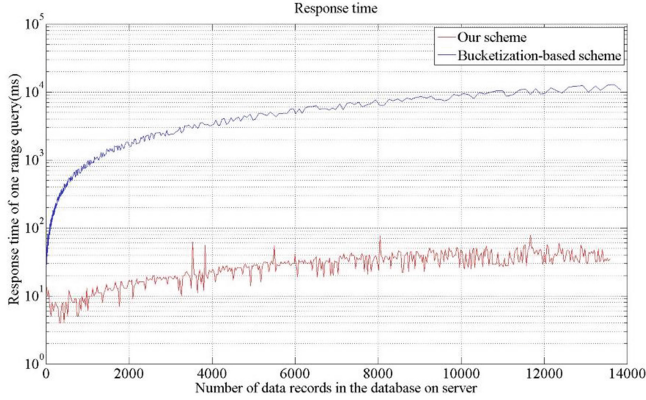


Fig. 3. Response time of our scheme and bucket system.

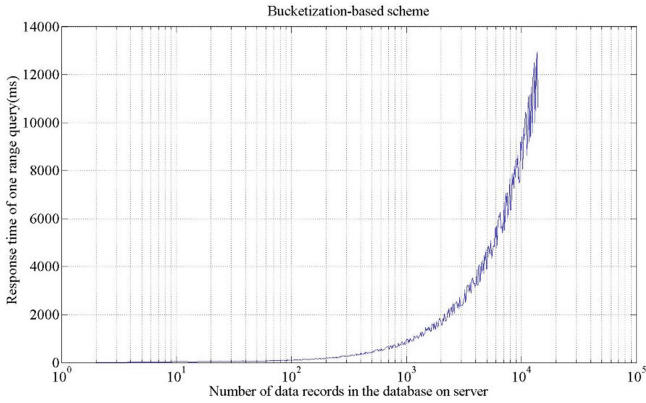


Fig. 4. Response time of the bucketization system when the data records increase.

Computation Cost of the Data Owner. We compare the computation time of the electricity company when the number of users and query dimension changes.

Figure 5 shows the computation time when the number of users in electricity company changes. From the two figures, it can illustrate the linear relationship when the users’ size increases no matter what the query dimension is. And from the results, we can see that our scheme incurs less computation cost than the Bucketization-based scheme when coping with large number of users.

In smart grid application, the number of users is very large. From the simulation results, we can estimate that our scheme operates well than the Bucketization-based scheme in smart grid. Therefore, our scheme is very suitable for large-scale smart grid systems.

Figure 6 describes the computation cost of the electricity company with fixed users versus the number of changing query dimension. It is easy to find that our

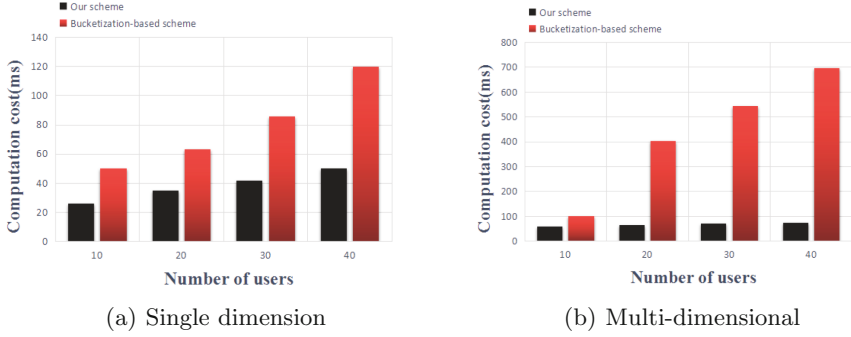


Fig. 5. The computation time of the electricity company when the number of users changes.

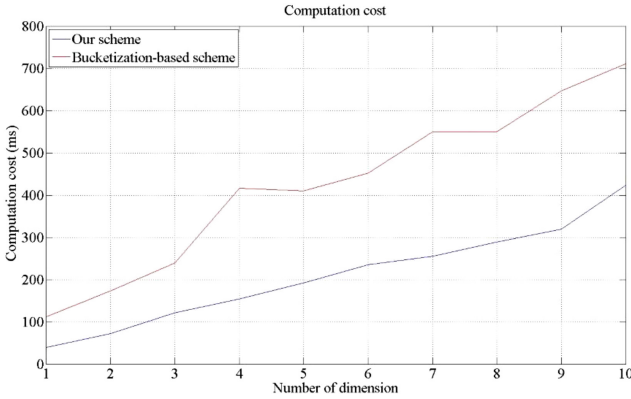
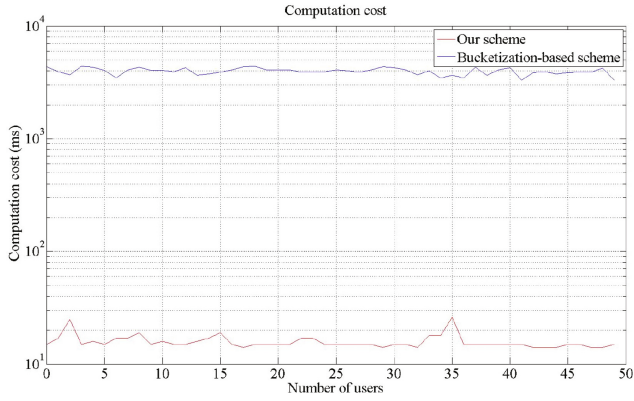


Fig. 6. The computation cost of the electricity company with fixed users versus the number of changing dimensions.

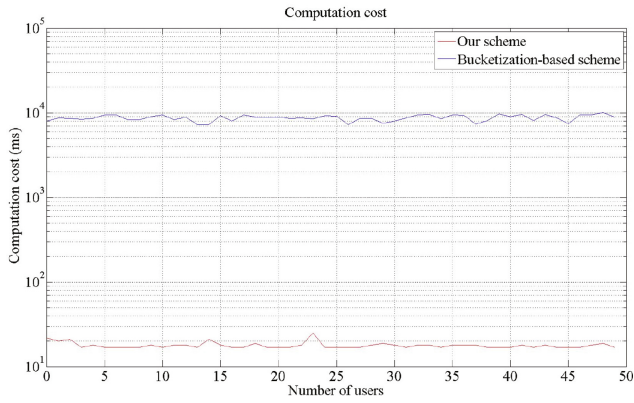
scheme incurs less computation cost than Bucketization-based scheme, especially when the query dimension is large in smart grid.

Computation Cost of Data Users. We compare the computation cost of the data users versus the users’ size in Fig. 7 and the number of query dimension in Fig. 8. From the figures, we can see that our scheme is always in lower computation cost no matter what the users’ size or the dimension is. Our scheme can greatly reduce the computation cost of data users, which is more important for data users in smart grid.

From the aforementioned analysis, We thus conclude that: (1) Our scheme can shorten the response time for a range query, which is significant for smart grid application. (2) As the users’ size and the query dimension increase, the computation cost of the electricity company in our scheme changes little, which is suitable for large-scale smart grid systems. (3) The computation cost in data



(a) Single dimension



(b) Multi-dimensional

Fig. 7. The computation time of the data users when the users connected to electricity company change.

users' size in our scheme always keep little. This is very important for the data user who need to audit much electricity usage data in real. Therefore, our scheme is efficient enough and suitable for smart grid application.

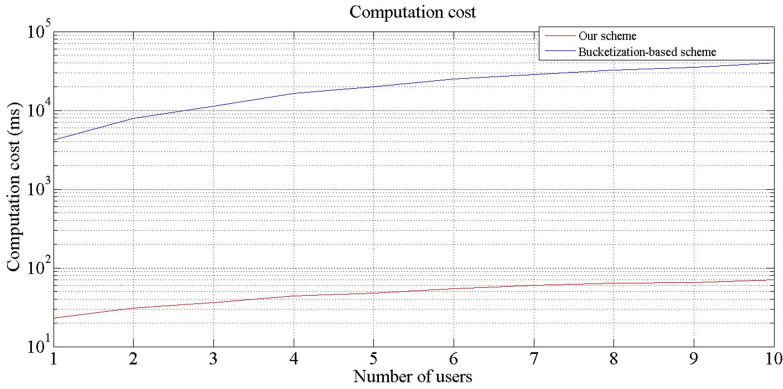


Fig. 8. The computation cost of the data user with fixed users versus the number of changing dimensions.

9 Conclusion

In this paper, we provide an efficient privacy-preserving scheme for range query in smart grid based on the modified Paillier cryptosystem. We achieved the range query in smart grid without disclosing the privacy of the electricity usage data and queries. The performance shows that our scheme can reduce the computation cost for both the data owner and data users, and shorten the response time of every range query, which is great significance for smart grid application.

Acknowledgments. This work was financially supported by National Natural Science Foundation of China with Grant No.61672195 and No. 61732022, National Key Research and Development Program of China with Grant No. 2016YFB0800804 and No. 2017YFB0803002, and Shenzhen Science and Technology Plan with Grant No. JCYJ20160318094336513 and No. JCYJ20160318094101317.

References

1. Wen, M., Lu, R., Zhang, K., Lei, J., Liang, X., Shen, X.: PaRQ: a privacy-preserving range query scheme over encrypted metering data for smart grid. *IEEE Trans. Emerg. Top. Comput.* **1**(1), 178–191 (2013)
2. Lu, R., Liang, X., Li, X., Lin, X., Shen, X.: Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Trans. Parallel Distrib. Syst.* **23**(9), 1621–1631 (2012)
3. Liang, X., Li, X., Lu, R., Lin, X., Shen, X.: UDP: usage-based dynamic pricing with privacy preservation for smart grid. *IEEE Trans. Smart Grid* **4**(1), 141–150 (2013)
4. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) *EUROCRYPT 2004*. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30

5. Zhang, B., Zhang, F.: An efficient public key encryption with conjunctive-subset keywords search. *J. Netw. Comput. Appl.* **34**(1), 262–267 (2011)
6. Baek, J., Safavi-Naini, R., Susilo, W.: On the integration of public key data encryption and public key encryption with keyword search. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) *ISC 2006*. LNCS, vol. 4176, pp. 217–232. Springer, Heidelberg (2006). https://doi.org/10.1007/11836810_16
7. Liu, Q., Wang, G., Wu, J.: An efficient privacy preserving keyword search scheme in cloud computing. In: *2009 International Conference on Computational Science and Engineering, CSE 2009*, vol. 2, pp. 715–720. IEEE (2009)
8. Wen, M., Lu, R., Lei, J., Li, H., Liang, X., Sherman Shen, X.: SESA: an efficient searchable encryption scheme for auction in emerging smart grid marketing. *Secur. Commun. Netw.* **7**(1), 234–244 (2014)
9. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order preserving encryption for numeric data. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pp. 563–574. ACM (2004)
10. Boldyreva, A., Chenette, N., Lee, Y., O’Neill, A.: Order-preserving symmetric encryption. In: Joux, A. (ed.) *EUROCRYPT 2009*. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_13
11. Boldyreva, A., Chenette, N., O’Neill, A.: Order-preserving encryption revisited: improved security analysis and alternative solutions. In: Rogaway, P. (ed.) *CRYPTO 2011*. LNCS, vol. 6841, pp. 578–595. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_33
12. Boneh, D., Waters, B.: Conjunctive, Subset, and Range Queries on Encrypted Data. In: Vadhan, S.P. (ed.) *TCC 2007*. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_29
13. Shi, E., Bethencourt, J., Chan, T.H.H., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. In: *IEEE Symposium on Security and Privacy, 2007, SP 2007*, pp. 350–364. IEEE (2007)
14. Wang, B., Hou, Y., Li, M., Wang, H., Li, H.: Maple: scalable multi-dimensional range search over encrypted cloud data with tree-based index. In: *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, pp. 111–122. ACM (2014)
15. Lu, Y.: Privacy-preserving logarithmic-time search on encrypted data in cloud. In: *NDSS* (2012)
16. Wong, W.K., Cheung, D.W., Kao, B., Mamoulis, N.: Secure knn computation on encrypted databases. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, pp. 139–152. ACM (2009)
17. Wang, P., Ravishankar, C.V.: Secure and efficient range queries on outsourced databases using Rp-trees. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 314–325. IEEE (2013)
18. Chi, J., Hong, C., Zhang, M., Zhang, Z.: Privacy-enhancing range query processing over encrypted cloud databases. In: Wang, J., Cellary, W., Wang, D., Wang, H., Chen, S.-C., Li, T., Zhang, Y. (eds.) *WISE 2015*. LNCS, vol. 9419, pp. 63–77. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26187-4_5
19. Hacıgümüş, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL over encrypted data in the database-service-provider model. In: *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp. 216–227. ACM (2002)
20. Hore, B., Mehrotra, S., Tsudik, G.: A privacy-preserving index for range queries. In: *Thirtieth International Conference on Very Large Data Bases*, pp. 720–731 (2004)

21. Hore, B., Mehrotra, S., Canim, M., Kantarcioglu, M.: Secure multidimensional range queries over outsourced data. *VLDB J.* **21**(3), 333–358 (2012)
22. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) *EUROCRYPT 1999*. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16