IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 22, 2019, accepted November 8, 2019, date of publication November 19, 2019,
date of current version December 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954156

A Framework for Privacy-Preserving Multi-Party Skyline Query Based on Homomorphic Encryption

MAHBOOB QAOSAR[1,2], **KAZI MD. ROKIBUL ALAM**[1,3], **ASIF ZAMAN**[2], **CHEN LI**[1], **SALEH AHMED**[4], **MD. ANISUZZAMAN SIDDIQUE**[2], **AND YASUHIKO MORIMOTO**[1]

[1]Graduate School of Engineering, Hiroshima University, Hiroshima 739-8527, Japan
[2]Department of Computer Science and Engineering, University of Rajshahi, Rajshahi 6205, Bangladesh
[3]Department of Computer Science and Engineering, Khulna University of Engineering and Technology, Khulna 9203, Bangladesh
[4]Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj 8100, Bangladesh

Corresponding author: Mahboob Qaosar (d172517@hiroshima-u.ac.jp)

This work was supported by KAKENHI, Japan, under Grant 16K00155 and Grant 17H01823. The work of M. Qaosar, A. Zaman, S. Ahmed, and M. A. Siddique was supported by the Japanese Government MEXT Scholarship.

ABSTRACT Nowadays, the management and analyses of 'big data' are becoming indispensable for numerous organizations all over the world. In many cases, multiple organizations want to perform data analyses on their combined databases. Skyline query is one of the popular operations for selecting representative objects from a large database, where any other object within the database does not dominate each of the representative objects, called 'skyline'. Like other data analytics operations, the multi-party skyline query can provide benefits to the participating organizations by retrieving the skyline objects from their combined databases. Such a multi-party skyline query demands the disclosure of individual parties' objects to others during the computation. But, owing to the data privacy and security concern of the present IT era, such disclosure of the individual parties' databases is strictly prohibited. Considering this issue, we are proposing a new framework for the privacy-preserving multi-party skyline query, exploiting additive homomorphic encryption along with data anonymization, perturbation, and randomization techniques. The underlying protocols within our proposed framework ensure that every participating party can identify its multi-party skyline objects without revealing the objects to others during the multi-party skyline query. The detailed privacy and security analyses show that the proposed framework can achieve the desired computation goal without privacy leakage. Besides, the performance evaluation through complexity analyses, extensive simulations, and comprehensive comparison also demonstrate the utility and the efficiency of the proposed framework.

INDEX TERMS Data mining, skyline query, multi-party computation, data privacy, Paillier cryptosystem, homomorphic encryption.

I. INTRODUCTION

Organizations throughout the world are producing a vast amount of data, known as 'big data'. Consequently, the demand for big data analytics tools is growing rapidly. These tools have attracted massive attention to organizations and researchers for making strategic decisions and for new knowledge acquisitions. Open market product pricing, risk management in investment, consumer buying pattern analysis, financial transaction analysis, health data analysis, etc. are remarkable examples of big data analyses. Still, big data is introducing new challenges for collection, storage, process, analyze, etc.

In the current trend of IT, multiple organizations dealing with similar kind of services are collecting compatible big data, and have noticed the importance of analytical results that can be found from the union of their databases. Such sort of joint data analyses requires multi-party computation over the combined databases of all organizations. Since many organizational databases may contain various sensitive data like personal or financial data, revealing these data can seriously violate the individuals' privacy and can be the reason of significant financial and goodwill loss for the organizations. As a result, when multiple organizations want to analyze

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

	Distance	Price
A	4	18
B	8	15
C	10	17
D	11	10
E	13	15
F	15	12
G	16	5
H	18	9

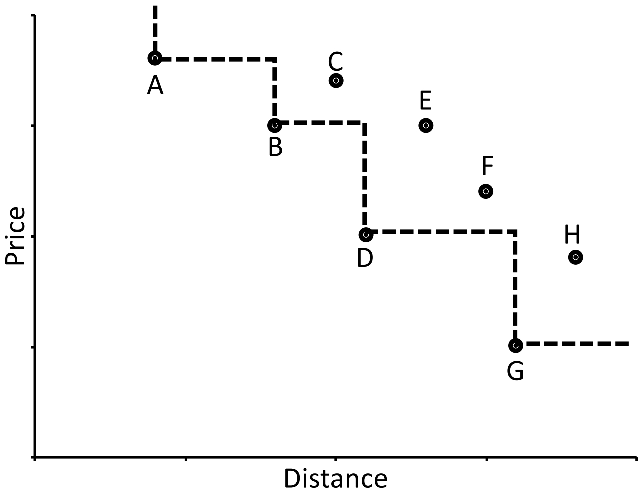

FIGURE 1. A skyline example of real estate property records.

their data jointly, no organization is willing to disclose their sensitive data. In this paper, we address this problem.

Already, the skyline query has gained popularity for selecting representative objects from a large database. It chooses a set of representative objects in such a way, in which no other object in the database dominates these representative objects [1], [2]. For example, the table in Fig. 1 shows a database of real estates, which contains records about real estates' price and its distance from the beach. From its corresponding plot diagram, points A, B, D, and G are not dominated by any other point in the database, and therefore, these are the skyline points. When a customer asks the agent to recommend some real estates, the agent can suggest some potential real estates from the skyline query.

Undoubtedly, the multi-party skyline query provides more benefits to the organizations by selecting skyline objects from the union of their databases. The organizations may want to locate their skyline objects that are not dominated by any other object of their combined databases. However, such computation is very sensitive *w.r.t.* security and privacy challenges. Let us assume that several agents have a collection of similar kind of real estate records. To provide better and competitive suggestions for their customers, every agent may also want to determine its real estates which are not dominated by any real estate of other agents. In such a case, all agents need to perform the multi-party skyline query on their union databases. The agents may also utilize the multi-party skyline query to assess their real estates. In conventional skyline query, it is not possible to obtain the multi-party skyline result without disclosing the objects to others.

Unfortunately, very few existing works [3]–[7] addressed the issues of data privacy and security for skyline query. However, their secure computation circumstances are different from our proposal. Besides, most of the existing works incorporate one or more semi-honest third-parties to conduct the privacy-preserving skyline query, where the privacy of the individual party's database profoundly depend on the credibility of the third party(s). Since the third party(s) may involve in the conspiracy, it is challenging to assume an unbiased third-party(s) who will be trusted by all parties.

Therefore, we propose a new framework for privacy-preserving multi-party skyline query in this paper. In the proposed framework, individual parties/organizations do not need to reveal their private databases to others. We design three intra-dependent protocols to implement the framework through which only the objects owner can identify its multi-party skyline objects. Even, no party is able to know the number of multi-party skyline objects of other parties.

The remaining part of this paper is organized as follows: we review some existing works on skyline query, privacy-preserving multi-party computation, and privacy-preserving skyline query in Section II. In Section III, we discuss the preliminaries of the skyline query and the Paillier cryptosystem, as required to develop the framework. Then, we explain the proposed system model with desired privacy properties in Section IV. In Section V, we specify the detail framework with brief explanations and examples. Next, we explain the privacy and security analyses for the proposed framework in Section VI. After that, we evaluate its performance, and compare with other work in Section VII. Finally, we conclude the paper in Section VIII.

II. RELATED WORKS

The works on skyline query processing, privacy-preserving multi-party computation, and privacy-preserving skyline query are related to this research work. Subsection II-A focuses on the skyline query, subsection II-B discusses about privacy-preserving multi-party computation, and subsection II-C highlights on privacy-preserving skyline query.

A. SKYLINE QUERY

Borzsonyi *et al.* [1], who are the introducer of the skyline operator, proposed three algorithms for computing skyline: *Block-Nested-Loops (BNL)*, *Divide-and-Conquer (D&C)*, and B-tree-based schemes. Later, Kossmann *et al.* [8] improved the *D&C* algorithm and proposed the *nearest-neighbor (NN)* algorithm for efficiently pruning out dominated objects by iteratively partitioning the data space based on the nearest objects. Similarly, Chomicki *et al.* [9] improved the *BNL* algorithm by presorting the objects. They proposed *Sort-Filter-Skyline (SFS)* as a variant of *BNL*. Another efficient algorithm for skyline query is *Branch-and-Bound Skyline (BBS)*, proposed by Papadias *et al.* [10], which is a progressive algorithm based on the *Best First Nearest Neighbor (BF-NN)* algorithm.

Besides, Balke *et al.* [11] introduced skyline queries in distributed environments. They proposed various approaches for computing distributed skyline efficiently from the vertically partitioned web information. On the other hand, both Wang *et al.* [12] and Chen *et al.* [13] proposed efficient skyline query processing frameworks in the structured P2P networks. Rocha *et al.* [14] also proposed a grid-based approach for distributed skyline processing. This approach assumed that each peer maintained a grid-based data summary structure for describing its data distribution.

B. PRIVACY-PRESERVING MULTI-PARTY COMPUTATION

Privacy-preserving multi-party computation is vital for modern business data processing. Yao [15] first introduced it for two-party setting and then Goldreich *et al.* [16] extended it for multi-party setting. According to [16], security in multi-party computation means that the individual parties' data remain secret during the computation and the parties could only get the computed results. Generally, the secure multi-party computation protocols are complicated than the specific purpose protocols.

Privacy-preserving multi-party data-mining problem is another example of secure multi-party computation. It is considered as one of the key research areas of 'big data'. Lindell and Pinkas [17] proposed the algorithm for performing the data mining operation on the combined databases of two parties, where one party does not disclose its database to another party during computation. In Agrawal's paper [18], the privacy-preserving data mining problem is described considering two parties: *Alice* and *Bob*; where *Alice* is allowed to conduct data mining operation on a private database owned by *Bob*, but *Bob* wants to prevent *Alice* from accessing precise information in individual data records, allowing *Alice* to conduct the data mining operations. Although these two problems are quite similar, the approaches proposed in [17] and [18] are different. Reference [17] used the secure multi-party computation protocols; while [18] applied the data perturbation method.

Most of the existing solutions for multi-party computation [19], [20] utilized homomorphic encryption for comparing the private data of the individual parties, although these protocols are highly expensive *w.r.t.* computation and communication complexity [21]. Lin and Tzeng [22] introduced another secure comparison protocol based on homomorphic encryption known as the 0-encoding and 1-encoding scheme. It is a two-party secure comparison protocol for comparing two private data in two rounds of data exchange. However, the complexity of the 0-encoding and 1-encoding scheme also depends on the length of the integer attribute value in the number of binary bits like secure comparison protocol proposed by Lin and Jaromczyk [19] and Veugen *et al.* [20].

The local differential privacy (LDP) schemes have been proposed for privacy-preserving distributed data collection [23]–[26]. By using the LDP schemes, the database owners can ensure the confidentiality of individual database records shared with a collector, while the collector computes on the differential private version of the database records to publish the statistical aggregate results from the collected databases. These schemes utilize various data anonymization techniques to maintain data privacy and also estimates the tradeoff between data utility and data privacy.

C. PRIVACY-PRESERVING SKYLINE QUERY

Like other privacy-preserving multi-party computation problems, the privacy-preserving multi-party skyline query is also being researched considering various perspectives.

Concerning the privacy of user's dynamic skyline query, three different frameworks were proposed by Chen *et al.* [4], Liu *et al.* [5], and Hua *et al.* [6]. Unfortunately, their secure skyline computation objectives are different from our current scenario. Within their frameworks, the data provider cannot know the user's dynamic skyline query. On the other hand, the user cannot know the entire private database of the data provider other than the skyline query result.

Liu *et al.* [7] proposed a skyline computation framework for two parties, which can also be deployable in a multi-party computation platform. They considered pruning out the dominated objects iteratively by using secure dominance comparison between two individual parties' objects. Although their proposed Efficient Secure Vector Comparison (ESVC) protocol between two parties does not disclose the object's attributes to one another, it reveals the dominance relation between their two specific objects, to both parties.

The framework proposed by Zaman *et al.* [3] transforms the multi-party objects' attributes securely into the rank of the attribute value on each dimension and then uses the attributes' rank for computing the multi-party skyline. Although it seemed to be efficient than relevant frameworks, still there exists suspicion that whither the attributes' rank of the objects can ensure the privacy adequately, or not.

III. PRELIMINARIES

This section defines the essential preliminaries considered for the proposed framework. Besides, the common notations used in this paper are introduced in Table 1.

A. DOMINANCE AND SKYLINE

Given a dataset $\boldsymbol{DS}$ with D-dimensions $\{d_1, d_2, \cdots, d_D\}$ and N objects $\{\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_N\}$, where $\boldsymbol{O}_i.d_j$ denotes the j-th dimension value of object $\boldsymbol{O}_i$. We assume that the smaller value in each dimension is better, without loss of generality.

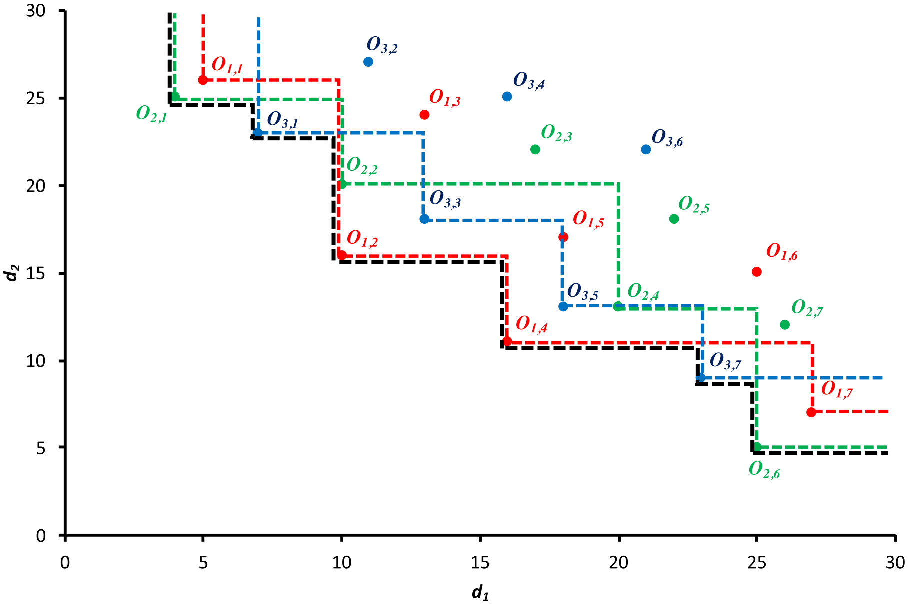

FIGURE 2. A skyline example with skyline additivity property.

TABLE 1. The summary of notations.

Notation	Definition
D	Object Dimension
$\boldsymbol{O}, \boldsymbol{A}, \boldsymbol{B}, \cdots$	Object/Vector/Array
$\boldsymbol{A} \prec \boldsymbol{B}$	Dominance relationship between $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{A}\|\boldsymbol{B}$	Concatenation of vectors $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{Party}_K$	K^{th} participating party
$Sky\left(\boldsymbol{DS}_K\right)$	Skyline objects of $\boldsymbol{Party}_K$
pk_K, sk_K	Public encryption key, private decryption key of $\boldsymbol{Party}_K$
$\mathbb{Z}$	Universal set of integer numbers
$\hat{+}, \hat{-}, \hat{\times}$	Homomorphic Addition, Subtraction, Multiplication
$[\boldsymbol{O}]_K$	$\boldsymbol{O}$ is encrypted by the public key pk_K of $\boldsymbol{Party}_K$;
$dcount^K_{L,i}$	The number of objects of $Sky\left(\boldsymbol{DS}_K\right)$, which dominate the i^{th} object of $Sky\left(\boldsymbol{DS}_L\right)$
$\pi(\cdot), \pi'(\cdot)$	Random permutation function

1) DOMINANCE

An object $\boldsymbol{O}_i \in DS$ is said to dominate another object $\boldsymbol{O}_j \in DS$, denoted as $\boldsymbol{O}_i \prec \boldsymbol{O}_j$, if $\boldsymbol{O}_i.d_k \leq \boldsymbol{O}_j.d_k$ $(1 \leq k \leq D)$ for all dimensions and $\boldsymbol{O}_i.d_l < \boldsymbol{O}_j.d_l$ $(1 \leq l \leq D)$ for at least one dimension. We call such $\boldsymbol{O}_i$ as *dominant object* and such $\boldsymbol{O}_j$ as *dominated object* between $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$. For example, in Fig. 2 object $\boldsymbol{O}_{2,2}(10, 20)$ is dominated by object $\boldsymbol{O}_{1,2}(10, 16)$, since $\boldsymbol{O}_{1,2}.d_2(16) < \boldsymbol{O}_{2,2}.d_2(20)$, although $\boldsymbol{O}_{1,2}.d_1(10) = \boldsymbol{O}_{2,2}.d_1(10)$.

2) SKYLINE

An object $\boldsymbol{O}_i \in \boldsymbol{DS}$ is said to be a skyline object of $\boldsymbol{DS}$, if and only if there is no such object $\boldsymbol{O}_j \in \boldsymbol{DS}$ $(j \neq i)$ that dominates $\boldsymbol{O}_i$. The skyline of $\boldsymbol{DS}$, denoted as $Sky(\boldsymbol{DS})$, is the set of skyline objects in $\boldsymbol{DS}$. For the dataset plotted in Fig. 2, objects $\boldsymbol{O}_{2,1}, \boldsymbol{O}_{3,1}, \boldsymbol{O}_{1,2}, \boldsymbol{O}_{1,4}, \boldsymbol{O}_{3,7}, \boldsymbol{O}_{2,6}$ are not dominated by any other objects. Thus, skyline query retrieves $Sky(\boldsymbol{DS}) = \{\boldsymbol{O}_{2,1}, \boldsymbol{O}_{3,1}, \boldsymbol{O}_{1,2}, \boldsymbol{O}_{1,4}, \boldsymbol{O}_{3,7}, \boldsymbol{O}_{2,6}\}$.

3) ADDITIVITY OF SKYLINE COMPUTATION [27]

Given a dataset $\boldsymbol{DS}$ that is composed by union of K datasets such that $\boldsymbol{DS} = \boldsymbol{DS}_1 \cup \cdots \cup \boldsymbol{DS}_K$, the following equation holds:

$$Sky(\boldsymbol{DS}) = Sky(Sky(\boldsymbol{DS}_1) \cup \cdots \cup Sky(\boldsymbol{DS}_K))$$

This implies that each skyline object of $\boldsymbol{DS}$ must be a skyline object of $\boldsymbol{DS}$'s subset. In Fig. 2, we denote that the red, the green, and the blue points represent the objects of $\boldsymbol{DS}_1$, $\boldsymbol{DS}_2$, and $\boldsymbol{DS}_3$, respectively. The skyline objects of $\boldsymbol{DS}_1$, $\boldsymbol{DS}_2$ and $\boldsymbol{DS}_3$ is given as $Sky(\boldsymbol{DS}_1) = \{\boldsymbol{O}_{1,1}, \boldsymbol{O}_{1,2}, \boldsymbol{O}_{1,4}, \boldsymbol{O}_{1,7}\}$, $Sky(\boldsymbol{DS}_2) = \{\boldsymbol{O}_{2,1}, \boldsymbol{O}_{2,2}, \boldsymbol{O}_{2,4}, \boldsymbol{O}_{2,6}\}$, and $Sky(\boldsymbol{DS}_3) = \{\boldsymbol{O}_{3,1}, \boldsymbol{O}_{3,3}, \boldsymbol{O}_{3,5}, \boldsymbol{O}_{3,7}\}$. It is apparent that the common skyline objects is given as $Sky(\boldsymbol{DS}) = \{\boldsymbol{O}_{2,1}, \boldsymbol{O}_{3,1}, \boldsymbol{O}_{1,2}, \boldsymbol{O}_{1,4}, \boldsymbol{O}_{3,7}, \boldsymbol{O}_{2,6}\}$, where $\{\boldsymbol{O}_{1,2}, \boldsymbol{O}_{1,4}\} \in Sky(\boldsymbol{DS}_1)$, $\{\boldsymbol{O}_{2,1}, \boldsymbol{O}_{2,6}\} \in Sky(\boldsymbol{DS}_2)$, and $\{\boldsymbol{O}_{3,1}, \boldsymbol{O}_{3,7}\} \in Sky(\boldsymbol{DS}_3)$.

We also introduce and frequently used two common terminologies throughout the paper: the *local skyline object* and the *global skyline object*. Here the local skyline object denotes the non-dominated skyline object of a sub-dataset, *i.e.*, an object of $Sky(\boldsymbol{DS}_1)$, while the global skyline object denotes the skyline object computed from the union of subdatasets, *i.e.*, an object of $Sky(\boldsymbol{DS})$.

B. PAILLIER CRYPTOSYSTEM

Paillier cryptosystem [28] is a probabilistic asymmetric encryption scheme that possesses additive homomorphic property. Consider pk and sk be the Paillier public encryption key and the private decryption key, respectively. Also assume m_1 and m_2 be two distinct plaintext integers while $[m_1]_{pk}$ and $[m_2]_{pk}$ represent their ciphertext, respectively. Based on this scenario, Paillier cryptosystem has the following additive homomorphic properties:

TABLE 2. Local datasets of the individual parties.

$\boldsymbol{DS}_1$ of $\boldsymbol{Party}_1$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{1,1}$	5	26
$\boldsymbol{O}_{1,2}$	10	16
$\boldsymbol{O}_{1,3}$	13	24
$\boldsymbol{O}_{1,4}$	16	11
$\boldsymbol{O}_{1,5}$	18	17
$\boldsymbol{O}_{1,6}$	25	15
$\boldsymbol{O}_{1,7}$	27	7

$\boldsymbol{DS}_2$ of $\boldsymbol{Party}_2$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{2,1}$	4	25
$\boldsymbol{O}_{2,2}$	10	20
$\boldsymbol{O}_{2,3}$	17	22
$\boldsymbol{O}_{2,4}$	20	13
$\boldsymbol{O}_{2,5}$	22	18
$\boldsymbol{O}_{2,6}$	25	5
$\boldsymbol{O}_{2,7}$	26	12

$\boldsymbol{DS}_3$ of $\boldsymbol{Party}_3$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{3,1}$	7	23
$\boldsymbol{O}_{3,2}$	11	27
$\boldsymbol{O}_{3,3}$	13	18
$\boldsymbol{O}_{3,4}$	16	25
$\boldsymbol{O}_{3,5}$	18	13
$\boldsymbol{O}_{3,6}$	21	22
$\boldsymbol{O}_{3,7}$	23	9

TABLE 3. Local skyline objects of the individual parties.

$Sky\,(\boldsymbol{DS}_1)$ of $\boldsymbol{Party}_1$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{1,1}$	5	26
$\boldsymbol{O}_{1,2}$	10	16
$\boldsymbol{O}_{1,4}$	16	11
$\boldsymbol{O}_{1,7}$	27	7

$Sky\,(\boldsymbol{DS}_2)$ of $\boldsymbol{Party}_2$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{2,1}$	4	25
$\boldsymbol{O}_{2,2}$	10	20
$\boldsymbol{O}_{2,4}$	20	13
$\boldsymbol{O}_{2,6}$	25	5

$Sky\,(\boldsymbol{DS}_3)$ of $\boldsymbol{Party}_3$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{3,1}$	7	23
$\boldsymbol{O}_{3,3}$	13	18
$\boldsymbol{O}_{3,5}$	18	13
$\boldsymbol{O}_{3,7}$	23	9

TABLE 4. Global skyline objects (GSO) of the individual parties.

GSO of $\boldsymbol{Party}_1$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{1,2}$	10	16
$\boldsymbol{O}_{1,4}$	16	11

GSO of $\boldsymbol{Party}_2$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{2,1}$	4	25
$\boldsymbol{O}_{2,6}$	25	5

GSO of $\boldsymbol{Party}_3$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{O}_{3,1}$	7	23
$\boldsymbol{O}_{3,7}$	23	9

- Homomorphic Addition

$$[m_1 + m_2]_{pk} := \left([m_1]_{pk} \times [m_2]_{pk}\right) \bmod n^2$$

- Homomorphic Multiplication

$$[k \times m_1]_{pk} := \left([m_1]_{pk}\right)^k \bmod n^2$$

where n is the part of Paillier public key and k is a constant integer.

IV. SYSTEM MODEL AND DESIRED PRIVACY PROPERTIES

In this section, we formalize the system model, and the desired privacy for our proposed framework.

A. SYSTEM MODEL

During the system design phase, we mainly concentrate on the privacy-aware multi-party skyline. We consider each party has a private dataset, where all parties are connected with each other. Without revealing the dataset to others, each party wants to identify the global skyline objects from their datasets that are not dominated by any object of their combined datasets. Here, we adopt the semi-honest adversary model and assume that all parties are honest-but-curious, *i.e.*, all parties strictly follow the protocol but intend to extract the private data of other parties from the computation.

Due to the additivity property of skyline computation, we can say that each object of the global skyline must be an object of any of the local skyline of the parties. Therefore, we assume that, before computing the global skyline securely, every party computes its local skyline objects. The local skyline computation can reduce the complexity of the global skyline computation significantly by pruning out the dominated objects from the local databases, and thus improve the computation efficiency.

Assume, Table 2 represents the private datasets of three individual parties, while Table 3 shows their local skyline objects. After computing the local skyline, each party wants to identify their global skyline objects without revealing their local skyline objects to others. Based on Table 3, Table 4 derives the global skyline objects owned by individual parties.

B. DESIRED PRIVACY

Our framework implicitly assumes that all participating parties do not collude with each other. It does not create any significant security threat for the honest parties even if some dishonest parties make any conspiracy. The proposed framework will possess the following privacy requirements:

- Any party does not expose its private objects directly to others during the computation. The parties either encrypt or anonymize the data before sharing it to others.
- Each party can only identify its own global skyline objects. No party is able to locate the global skyline objects of other parties; even a party cannot know how many global skyline objects are owned by other parties.

For example, after secure comparison between the local skyline objects of Table 3, $\boldsymbol{Party}_1$ has no information about a global skyline object that is owned by $\boldsymbol{Party}_2$ or $\boldsymbol{Party}_3$. Also, $\boldsymbol{Party}_1$ cannot know how many global objects $\boldsymbol{Party}_2$ and $\boldsymbol{Party}_3$ have owned.

- Any party cannot know whether its global skyline object dominates any object of others or not. After computation, each party can locate its own global skyline objects, but any party cannot know whether its global skyline objects dominate any objects or not. According to Table 4, after secure computation, $\boldsymbol{Party}_1$ can identify that $\boldsymbol{O_{1,2}}$ is a global skyline object, but $\boldsymbol{Party}_1$ cannot know whether $\boldsymbol{O_{1,2}}$ dominates any local skyline object of others, or not.
- Any party cannot know how many objects of others dominate its dominated objects. If a local skyline object is not a global skyline object, it is evident that at least an object of other parties dominates a specific dominated object, but any party cannot know the number of dominant objects for a specific dominated object precisely. According to Table 3 and Table 4, $\boldsymbol{Party}_2$ can determine $\boldsymbol{O_{2,4}}$ is not a global skyline object, but $\boldsymbol{Party}_2$ cannot know how many local skyline objects of $\boldsymbol{Party}_1$ or $\boldsymbol{Party}_3$ dominates $\boldsymbol{O_{2,4}}$.
- When the number of parties is more than two, no party can identify any particular party, whose object(s) dominates its specific dominated object. Using secure computation with $\boldsymbol{Party}_1$ and $\boldsymbol{Party}_2$, $\boldsymbol{Party}_3$ can find that $\boldsymbol{O_{3,3}}$ is dominated by other parties' object(s). However, $\boldsymbol{Party}_3$ is unable to know: $\boldsymbol{O_{3,3}}$ is dominated by $\boldsymbol{Party}_1$'s object(s), or $\boldsymbol{Party}_2$'s object(s), or both parties' objects.

V. PROPOSED FRAMEWORK

Initially, each party computes its local skyline objects, generates the key pair of Paillier cryptosystem, and distributes the public encryption key to others prior to multi-party skyline computation. The detail of Paillier cryptosystem and its homomorphic properties are available in [28]. We design three intra-dependent protocols to build our proposed framework. These are: the Multi-Party Skyline (MPS) protocol, the Dominant Objects Counter (DOC) protocol, and the Secure Dominance Comparison (SDC) protocol. In our framework, the MPS protocol applies the DOC protocol among every pair of parties; whereas the DOC protocol utilizes the SDC protocol to compare the dominance relationships among every pair of individual parties' two local skyline objects. Now, we describe the DOC, the SDC, and the MPS protocols in subsections V-A, V-B, and V-C, respectively.

A. DOMINANT OBJECTS COUNTER (DOC) PROTOCOL

The DOC protocol is a two-party protocol. For each local skyline object of both parties, the DOC protocol securely counts the dominant objects within the opposite party's local skyline objects. Suppose, $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$ are two parties. $\boldsymbol{Party}_A$ has $Sky\left(\boldsymbol{DS}_A\right)$, and $\boldsymbol{Party}_B$ has $Sky\left(\boldsymbol{DS}_B\right)$ as their local skyline objects. Furthermore, $\boldsymbol{Party}_A$ has (pk_A, sk_A), and $\boldsymbol{Party}_B$ has (pk_B, sk_B) as their key pairs. Algorithm 1 briefly describes the DOC protocol, and Fig. 3 depicts its data-flow diagram.

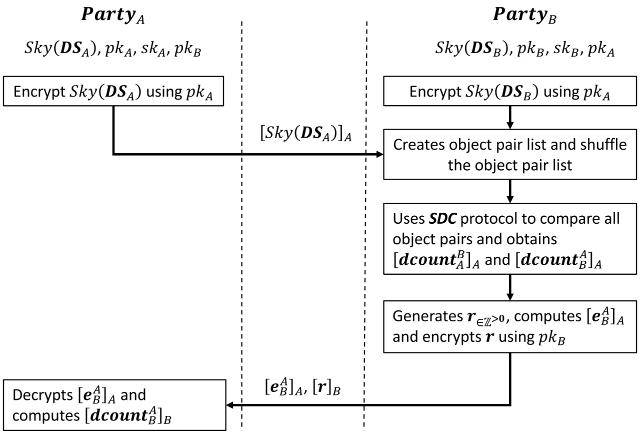
$\boldsymbol{Party}_A$
$Sky(\boldsymbol{DS}_A), pk_A, sk_A, pk_B$
Encrypt $Sky(\boldsymbol{DS}_A)$ using pk_A
$[Sky(\boldsymbol{DS}_A)]_A$
Decrypts $[\boldsymbol{e}_B^A]_A$ and computes $[\boldsymbol{dcount}_B^A]_B$
$[\boldsymbol{e}_B^A]_A$, $[\boldsymbol{r}]_B$
$\boldsymbol{Party}_B$
$Sky(\boldsymbol{DS}_B), pk_B, sk_B, pk_A$
Encrypt $Sky(\boldsymbol{DS}_B)$ using pk_A
Creates object pair list and shuffle the object pair list
Uses ***SDC*** protocol to compare all object pairs and obtains $[\boldsymbol{dcount}_A^B]_A$ and $[\boldsymbol{dcount}_B^A]_A$
Generates $\boldsymbol{r}_{\in\mathbb{Z}>0}$, computes $[\boldsymbol{e}_B^A]_A$ and encrypts $\boldsymbol{r}$ using pk_B

FIGURE 3. Data-flow diagram of the DOC protocol.

At the beginning of this protocol, $\boldsymbol{Party}_A$ encrypts $Sky\left(\boldsymbol{DS}_A\right)$ using pk_A and sends $\left[Sky\left(\boldsymbol{DS}_A\right)\right]_A$ to $\boldsymbol{Party}_B$. $\boldsymbol{Party}_B$ also encrypts $Sky\left(\boldsymbol{DS}_B\right)$ using pk_A. After that, $\boldsymbol{Party}_B$ creates the encrypted dominant objects counter field $\left[dcount_{A,i}^{B}\right]_A$ and $\left[dcount_{B,j}^{A}\right]_A$ for each object $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ and $\boldsymbol{B}_{j\in Sky(\boldsymbol{DS}_B)}$, and assigns $[0]_A$ as the initial value of each dominant objects counter. Here $dcount_{A,i}^{B}$ counts the objects in $Sky\left(\boldsymbol{DS}_B\right)$, which dominates $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$. Similarly, $dcount_{B,j}^{A}$ counts the objects in $Sky\left(\boldsymbol{DS}_A\right)$, which dominates $\boldsymbol{B}_{j\in Sky(\boldsymbol{DS}_B)}$.

Next, $\boldsymbol{Party}_B$ creates an object pair list from the Cartesian product of $Sky\left(\boldsymbol{DS}_A\right)$ and $Sky\left(\boldsymbol{DS}_B\right)$, *i.e.*, $Sky\left(\boldsymbol{DS}_A\right) \times Sky\left(\boldsymbol{DS}_B\right) = \left\{\left(\boldsymbol{A}_i, \boldsymbol{B}_j\right) | \boldsymbol{A}_i \in Sky\left(\boldsymbol{DS}_A\right) \text{ and } \boldsymbol{B}_j \in Sky\left(\boldsymbol{DS}_B\right)\right\}$. Then shuffle the object pair list randomly so that the list does not follow any chronological sequence. After that, $\boldsymbol{Party}_B$ uses the SDC protocol to compare the dominance relation between each pair of objects from the shuffled list. $\boldsymbol{Party}_B$ also randomizes the parameter order of the SDC protocol according to Step 6 of Algorithm 1. Because of the random shuffling of the object pair list and the parameter order randomization, $\boldsymbol{Party}_A$ cannot distinguish the objects (even $\boldsymbol{Party}_A$'s own local skyline objects), which are being compared through the SDC protocol.

The two output values of the SDC protocol obtained by $\boldsymbol{Party}_B$ denote the dominance relation between the compared objects. Among these two objects, if an object dominates another object, the output of the SDC protocol for the dominated object will be 1, whereas it will be 0 for the dominant object. But, both outputs will be 0 if the compared objects do not dominate each other. Since $\boldsymbol{Party}_A$ encrypts the outputs of the SDC protocol by pk_A, $\boldsymbol{Party}_B$ cannot know the dominance relation between two specific objects. However, using homomorphic addition, $\boldsymbol{Party}_B$ can add the encrypted outputs of the SDC protocol with the associated encrypted

Algorithm 1 Dominant Objects Counter (DOC) protocol

```
Input:
    $\boldsymbol{Party}_A$ has $Sky\,(\boldsymbol{DS}_A)$, $pk_A$, $sk_A$ and $pk_B$;
    $\boldsymbol{Party}_B$ has $Sky\,(\boldsymbol{DS}_B)$, $pk_B$, $sk_B$, and $pk_A$;
Output:
    $\boldsymbol{Party}_A$ gets $\left[\boldsymbol{dcount}_B^A\right]_B$ for $Sky\,(\boldsymbol{DS}_B)$;
    $\boldsymbol{Party}_B$ gets $\left[\boldsymbol{dcount}_A^B\right]_A$ for $Sky\,(\boldsymbol{DS}_A)$;
    $\underline{\boldsymbol{Party}_A}$:
1:  Encrypts $Sky\,(\boldsymbol{DS}_A)$ using $pk_A$ and sends $[Sky\,(\boldsymbol{DS}_A)]_A$ to $\boldsymbol{Party}_B$;
    $\underline{\boldsymbol{Party}_B}$:
2:  Encrypts $Sky\,(\boldsymbol{DS}_B)$ using $pk_A$;
3:  Creates dominant objects counter array $\left[\boldsymbol{dcount}_A^B\right]_A$ and $\left[\boldsymbol{dcount}_B^A\right]_A$ and assign $[0]_A$ as initial value;
4:  Creates an object pair list from the Cartesian product $Sky\,(\boldsymbol{DS}_A) \times Sky\,(\boldsymbol{DS}_B)$, and randomly shuffle the object pair list;
5:  for all pair $\left(\left[\boldsymbol{A}_{i \in Sky(\boldsymbol{DS}_A)}\right]_A, \left[\boldsymbol{B}_{j \in Sky(\boldsymbol{DS}_B)}\right]_A\right)$ do
6:      Randomly computes either
            i. $\left(\left[dom_{A_i}\right]_A, \left[dom_{B_j}\right]_A\right) \leftarrow \boldsymbol{SDC}\left([\boldsymbol{A}_i]_A, \left[\boldsymbol{B}_j\right]_A\right)$ or
            ii. $\left(\left[dom_{B_j}\right]_A, \left[dom_{A_i}\right]_A\right) \leftarrow \boldsymbol{SDC}\left(\left[\boldsymbol{B}_j\right]_A, [\boldsymbol{A}_i]_A\right)$;
7:      Computes $\left[dcount_{A,i}^B\right]_A := \left[dcount_{A,i}^B\right]_A \hat{+} \left[dom_{A_i}\right]_A$;
8:      Computes $\left[dcount_{B,j}^A\right]_A := \left[dcount_{B,j}^A\right]_A \hat{+} \left[dom_{B_j}\right]_A$;
9:  end for
    ▷ $\left[\boldsymbol{dcount}_A^B\right]_A$ and $\left[\boldsymbol{dcount}_B^A\right]_A$ contain the number of dominant objects for $Sky\,(\boldsymbol{DS}_A)$ and $Sky\,(\boldsymbol{DS}_B)$
10: For $\boldsymbol{dcount}_B^A$, generates random integer array $\boldsymbol{r}_{\in \mathbb{Z}^{>0}}$, computes $\left[\boldsymbol{e}_B^A\right]_A := \left[\boldsymbol{dcount}_B^A\right]_A \hat{+} [\boldsymbol{r}]_A$,
    and encrypts $\boldsymbol{r}$ using $pk_B$ to obtain $[\boldsymbol{r}]_B$;
11: Sends $\left[\boldsymbol{e}_B^B\right]_A$ and $[\boldsymbol{r}]_B$ to $\boldsymbol{Party}_A$;
    $\underline{\boldsymbol{Party}_A}$:
12: Decrypts $\left[\boldsymbol{e}_B^A\right]_A$ using $sk_A$ and encrypts $\boldsymbol{e}_B^A$ using $pk_B$ to obtain $\left[\boldsymbol{e}_B^A\right]_B$;
13: Computes $\left[\boldsymbol{dcount}_B^A\right]_B := \left[\boldsymbol{e}_B^A\right]_B \hat{-} [\boldsymbol{r}]_B$;
```

dominant objects counters of the compared objects. In this purpose, $\boldsymbol{Party}_B$ applies Step 7 and Step 8 of Algorithm 1.

After comparing all pairs of objects following Step 5 to Step 9 of Algorithm 1, $\left[\boldsymbol{dcount}_A^B\right]_A$ holds the number of dominant objects in $Sky\,(\boldsymbol{DS}_B)$ for each object of $Sky\,(\boldsymbol{DS}_A)$. Also, $\left[\boldsymbol{dcount}_B^A\right]_A$ holds the number of dominant objects in $Sky\,(\boldsymbol{DS}_A)$ for each object of $Sky\,(\boldsymbol{DS}_B)$. Table 5 and Table 6 are considered as the examples of the 2-dimensional local skyline objects of $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$, respectively, and from these we can compare between their local skyline objects. Here we can observe that $\boldsymbol{A}_1$ and $\boldsymbol{A}_4$ are not dominated by any local skyline object of $\boldsymbol{Party}_B$. Similarly, $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ are not dominated by any local skyline object of $\boldsymbol{Party}_A$. On the other hand, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$, $\boldsymbol{B}_1$, and $\boldsymbol{B}_4$ are dominated objects, since $(\boldsymbol{A}_1 \prec \boldsymbol{B}_1)$, $(\{\boldsymbol{B}_2, \boldsymbol{B}_3\} \prec \boldsymbol{A_2})$, $(\boldsymbol{B}_3 \prec \boldsymbol{A_3})$, and $(\boldsymbol{A}_4 \prec \boldsymbol{B}_4)$. Therefore, after secure computation following Step 5 to Step 9 of Algorithm 1, $\boldsymbol{Party}_B$ gets $\left[\boldsymbol{dcount}_A^B\right]_A$ and $\left[\boldsymbol{dcount}_B^A\right]_A$ as shown in Table 7.

TABLE 5. $Sky\,(\boldsymbol{DS_A})$.

id	d_1	d_2
$\boldsymbol{A}_1$	3	8
$\boldsymbol{A}_2$	6	7
$\boldsymbol{A}_3$	7	5
$\boldsymbol{A}_4$	8	2

Since the skyline objects are not dominated by any object, the number of dominant objects of a skyline object is zero. However, $\boldsymbol{Party}_B$ is unable to differentiate the non-dominated objects since $\boldsymbol{dcount}_A^B$, and $\boldsymbol{dcount}_B^A$ are encrypted by pk_A. In contrast, $\boldsymbol{Party}_A$ also cannot determine the global skyline

TABLE 6. ***Sky*** $(\boldsymbol{DS_B})$.

id	d_1	d_2
$\boldsymbol{B}_1$	4	9
$\boldsymbol{B}_2$	5	6
$\boldsymbol{B}_3$	6	4
$\boldsymbol{B}_4$	9	3

objects, since $\boldsymbol{Party}_A$ does not have the dominant objects counter.

Now, $\boldsymbol{Party}_A$ has to get the number of its dominant objects for each local skyline objects of $\boldsymbol{Party}_B$ in encrypted form. Therefore, $\boldsymbol{Party}_B$ first generates random positive integer array $\boldsymbol{r}_{\in\mathbb{Z}^{>0}}$ and computes $\left[\boldsymbol{e}_B^A\right]_A := \left[\boldsymbol{dcount}_B^A\right]_A \hat{+} [\boldsymbol{r}]_A$ through homomorphic addition. $\boldsymbol{Party}_B$ also encrypts $\boldsymbol{r}$ using pk_B to obtain $[\boldsymbol{r}]_B$. After that, $\boldsymbol{Party}_B$ sends $\left[\boldsymbol{e}_B^A\right]_A$ and $[\boldsymbol{r}]_B$ to $\boldsymbol{Party}_A$.

TABLE 7. **Content of encrypted dominant objects counters obtained through Step 5 to Step 9 of Algorithm 1.**

id	$\left[\boldsymbol{dcount}_A^B\right]_A$
$\boldsymbol{A}_1$	$[0]_A$
$\boldsymbol{A}_2$	$[2]_A$
$\boldsymbol{A}_3$	$[1]_A$
$\boldsymbol{A}_4$	$[0]_A$

id	$\left[\boldsymbol{dcount}_B^A\right]_A$
$\boldsymbol{B}_1$	$[1]_A$
$\boldsymbol{B}_2$	$[0]_A$
$\boldsymbol{B}_3$	$[0]_A$
$\boldsymbol{B}_4$	$[1]_A$

Although $\boldsymbol{Party}_A$ can decrypt $\left[\boldsymbol{e}_B^A\right]_A$ using sk_A, it cannot know anything about the local skyline objects of $\boldsymbol{Party}_B$ from the decrypted value. However, $\boldsymbol{Party}_A$ can obtain $\left[\boldsymbol{dcount}_B^A\right]_B$ for *Sky* $(\boldsymbol{DS}_B)$ by computing through Step 12 and Step 13 of Algorithm 1. Using the column $\left[\boldsymbol{dcount}_B^A\right]_A$ of Table 7, Table 8 presents the computation results of Step 10 to Step 13 of Algorithm 1.

TABLE 8. **Example of Step 10 to Step 13 of Algorithm 1.**

	$\boldsymbol{Party}_B$			$\boldsymbol{Party}_A$			
id	$\left[\boldsymbol{dcount}_B^A\right]_A$	$\boldsymbol{r}$	$\left[\boldsymbol{e}_B^A\right]_A$	$\boldsymbol{e}_B^A$	$\left[\boldsymbol{e}_B^A\right]_B$	$[\boldsymbol{r}]_B$	$\left[\boldsymbol{dcount}_B^A\right]_B$
$\boldsymbol{B}_1$	$[1]_A$	5	$[6]_A$	6	$[6]_B$	$[5]_B$	$[1]_B$
$\boldsymbol{B}_2$	$[0]_A$	7	$[7]_A$	7	$[7]_B$	$[7]_B$	$[0]_B$
$\boldsymbol{B}_3$	$[0]_A$	4	$[4]_A$	4	$[4]_B$	$[4]_B$	$[0]_B$
$\boldsymbol{B}_4$	$[1]_A$	8	$[9]_A$	9	$[9]_B$	$[8]_B$	$[1]_B$

Within the MPS protocol, the encrypted dominant objects counter obtained by one party for each local skyline object of another party will be used for computing multi-party skyline, from which only the individual party can identify its global skyline objects.

B. SECURE DOMINANCE COMPARISON (SDC) PROTOCOL

The SDC protocol is a sub-protocol of the DOC protocol, and it is designed to compare the dominance relation between two parties' encrypted objects. It is the principal component of the proposed framework. Same as the DOC protocol, we explain the SDC protocol considering two parties: $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$, where $\boldsymbol{Party}_A$ has the key pair (pk_A, sk_A), and $\boldsymbol{Party}_B$ has the public key pk_A and two encrypted objects $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$. Among these two encrypted objects, one object is owned by $\boldsymbol{Party}_A$, and another one is owned by $\boldsymbol{Party}_B$. As already described within the DOC protocol, $\boldsymbol{Party}_A$ can not know its particular object, which is compared through the SDC protocol.

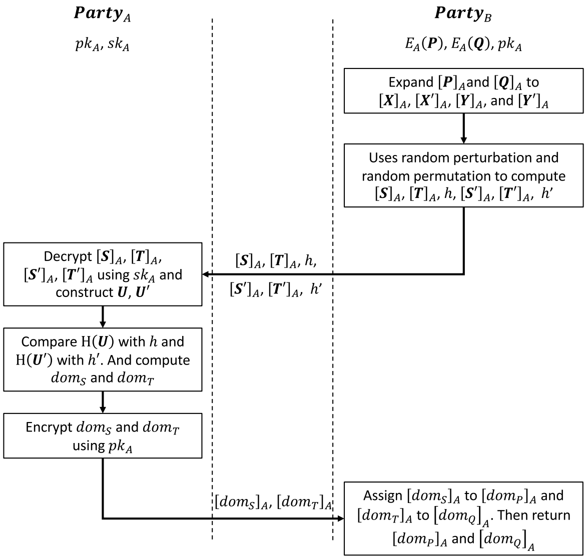
$Party_A$
pk_A, sk_A
$Party_B$
$E_A(\boldsymbol{P}), E_A(\boldsymbol{Q}), pk_A$
Expand $[\boldsymbol{P}]_A$and $[\boldsymbol{Q}]_A$ to $[\boldsymbol{X}]_A$, $[\boldsymbol{X}']_A$, $[\boldsymbol{Y}]_A$, and $[\boldsymbol{Y}']_A$
Uses random perturbation and random permutation to compute $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, h, $[\boldsymbol{S}']_A$, $[\boldsymbol{T}']_A$, h'
$[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, h,
$[\boldsymbol{S}']_A$, $[\boldsymbol{T}']_A$, h'
Decrypt $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $[\boldsymbol{S}']_A$, $[\boldsymbol{T}']_A$ using sk_A and construct $\boldsymbol{U}$, $\boldsymbol{U}'$
Compare $\mathrm{H}(\boldsymbol{U})$ with h and $\mathrm{H}(\boldsymbol{U}')$ with h'. And compute dom_S and dom_T
Encrypt dom_S and dom_T using pk_A
$[dom_S]_A$, $[dom_T]_A$
Assign $[dom_S]_A$ to $[dom_P]_A$ and $[dom_T]_A$ to $\left[dom_Q\right]_A$. Then return $[dom_P]_A$ and $\left[dom_Q\right]_A$

FIGURE 4. **Data-flow diagram of the SDC protocol.**

The SDC protocol assures that $\boldsymbol{Party}_A$ cannot know $\boldsymbol{Party}_B$'s object, whereas $\boldsymbol{Party}_B$ is unable to know the dominance relation between two specific objects. We design the SDC protocol obeying the basic principle of the ESVC Protocol [7]. However, to improve the computation efficiency, we ignore the 0-encoding and 1-encoding scheme based secure integer comparison protocol [22] used in the ESVC protocol. Instead, we adopt the data anonymization, perturbation, and randomization techniques. Furthermore, to maintain the desired privacy, we encrypt the dominance relation between two objects. Algorithm 2 describes the SDC protocol and Fig. 4 depicts its data-flow diagram.

We acknowledge three types of dominance relations between two objects $\boldsymbol{P}$ and $\boldsymbol{Q}$: either (1) $\boldsymbol{P} \prec \boldsymbol{Q}$, or (2) $\boldsymbol{Q} \prec \boldsymbol{P}$, or (3) $\boldsymbol{P}$ and $\boldsymbol{Q}$ do not dominate each other. To achieve the dominance relation between two objects, at first, $\boldsymbol{Party}_B$ expands D-dimensional encrypted objects $[\boldsymbol{P}\,(p_1, \cdots, p_D)]_A$ and $[\boldsymbol{Q}\,(q_1, \cdots, q_D)]_A$ into four $2D$ length encrypted vectors $[\boldsymbol{X}]_A$, $\left[\boldsymbol{X}'\right]_A$, $[\boldsymbol{Y}]_A$ and $\left[\boldsymbol{Y}'\right]_A$. In this regard, $\boldsymbol{Party}_B$ generates four $2D$ length random integer array to anonymize the vector elements using arbitrary transformation. These are $\boldsymbol{M} = (m_1, \ldots, m_{2D})_{\in\mathbb{Z}^{>1}}$, $\boldsymbol{M}' = (m_1', \ldots, m_{2D}')_{\in\mathbb{Z}^{>1}}$, $\boldsymbol{K} = (k_1, \ldots, k_{2D})_{\in\mathbb{Z}^{>0}}$, and $\boldsymbol{K}' = (k_1', \ldots, k_{2D}')_{\in\mathbb{Z}^{>0}}$.

Then, by applying the homomorphic addition and multiplication properties of Paillier cryptosystem, $\boldsymbol{Party}_B$ expands $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$ into $[\boldsymbol{X}]_A$, $\left[\boldsymbol{X}'\right]_A$, $[\boldsymbol{Y}]_A$, and $\left[\boldsymbol{Y}'\right]_A$ using the

Algorithm 2 Secure Dominance Comparison (SDC) Protocol

```
Input: $\boldsymbol{Party}_B$ has $[\boldsymbol{P}]_A$, $[\boldsymbol{Q}]_A$ and $pk_A$; $\boldsymbol{Party}_A$ has $sk_A$ and $pk_A$;
Output: $\boldsymbol{Party}_B$ gets $[dom_P]_A$ and $\left[dom_Q\right]_A$
    Party_B:
    Expands $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$ into four $2D$ length vector: $[\boldsymbol{X}]_A$, $\left[\boldsymbol{X'}\right]_A$, $[\boldsymbol{Y}]_A$ and $\left[\boldsymbol{Y'}\right]_A$;
    Constructs two $2D$ length binary vector: $\boldsymbol{V} = (1_1, ..., 1_D, 0_{D+1}, ..., 0_{2D})$ and $\boldsymbol{V'} = (1_1, ..., 1_D, 0_{D+1}, ..., 0_{2D})$;
    Generates two $2D$ length random binary vector: $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_{2D})_{\sigma_i \in 0,1}$ and $\boldsymbol{\sigma'} = (\sigma'_1, ..., \sigma'_{2D})_{\sigma'_i \in 0,1}$;
        i. Swaps each element $\left[x_{i_{\in X}}\right]_A$ and $\left[y_{i_{\in Y}}\right]_A$ if $\sigma_i = 1$;
        ii. Swaps each element $\left[x'_{i_{\in X'}}\right]_A$ and $\left[y'_{i_{\in Y'}}\right]_A$ if $\sigma'_i = 1$;
        iii. Computes $\boldsymbol{W} := \boldsymbol{V} \bigoplus \boldsymbol{\sigma}$ and $\boldsymbol{W'} := \boldsymbol{V'} \bigoplus \boldsymbol{\sigma'}$;
    i. Generates four $D$ length random positive integer vector: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\alpha'}$, and $\boldsymbol{\beta'}$;
        ii. Creates $D$ length binary vector $\boldsymbol{\rho}$ and set $\rho_{i_{\in\rho}} = 1$ if $\alpha_{i_{\in\alpha}} > \beta_{i_{\in\beta}}$ else set $\rho_{i_{\in\rho}} = 0$;
        iii. Creates $D$ length binary vector $\boldsymbol{\rho'}$ and set $\rho'_{i_{\in\rho'}} = 1$ if $\alpha'_{i_{\in\alpha'}} < \beta'_{i_{\in\beta'}}$ else set $\rho'_{i_{\in\rho'}} = 0$;
        iv. Encrypts $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\alpha'}$, and $\boldsymbol{\beta'}$ using $pk_A$;
    i. Computes $[\boldsymbol{S}]_A \leftarrow \pi\left(\left[\boldsymbol{X}|\boldsymbol{\alpha}\right]_A\right)$, $[\boldsymbol{T}]_A \leftarrow \pi\left(\left[\boldsymbol{Y}|\boldsymbol{\beta}\right]_A\right)$, and $\boldsymbol{G} \leftarrow \pi\left(\boldsymbol{W}|\boldsymbol{\rho}\right)$;
        ii. Computes $\left[\boldsymbol{S'}\right]_A \leftarrow \pi'\left(\left[\boldsymbol{X'}|\boldsymbol{\alpha'}\right]_A\right)$, $\left[\boldsymbol{T'}\right]_A \leftarrow \pi'\left(\left[\boldsymbol{Y'}|\boldsymbol{\beta'}\right]_A\right)$, and $\boldsymbol{G'} \leftarrow \pi'\left(\boldsymbol{W'}|\boldsymbol{\rho'}\right)$;;
    Uses hash function to compute $h := H(\boldsymbol{G})$ and $h' := H(\boldsymbol{G'})$;
    Sends $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $h$, $\left[\boldsymbol{S'}\right]_A$, $\left[\boldsymbol{T'}\right]_A$, and $h'$ to Party_A;
    Party_A:
    Decrypts $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $\left[\boldsymbol{S'}\right]_A$, and $\left[\boldsymbol{T'}\right]_A$ using private decryption key $sk_A$;
    Constructs two $3D$ length binary vector $\boldsymbol{U} = (u_1, ..., u_{3D})$ and $\boldsymbol{U'} = (u'_1, ..., u'_{3D})$;
        i. if $t_{i_{\in T}} > s_{i_{\in S}}$ then set $u_i := 1$ else set $u_i := 0$;
        ii. if $t'_{i_{\in T'}} < s'_{i_{\in S'}}$ then set $u'_i := 1$ else set $u'_i := 0$;
    if $H(\boldsymbol{U}) = h$ and $H(\boldsymbol{U'}) \neq h'$ then set $dom_S := 1, dom_T := 0$;        ▷ $[\boldsymbol{T} \prec \boldsymbol{S}]$
    else if $H(\boldsymbol{U}) \neq h$ and $H(\boldsymbol{U'}) = h'$ then set $dom_S := 0, dom_T := 1$;   ▷ $[\boldsymbol{S} \prec \boldsymbol{T}]$
    else set $dom_S := 0, dom_T := 0$;        ▷ [$\boldsymbol{S}$ and $\boldsymbol{T}$ do not dominate each other]
    end if
    Sends $[dom_S]_A$ and $[dom_T]_A$ to Party_B;
    Party_B:
    Assigns $[dom_P]_A := [dom_S]_A$ and $\left[dom_Q\right]_A := [dom_T]_A$;
```

following equations:

- $[x_i]_A := (2m_i \hat{\times} [p_i]_A) \hat{+} [k_i + m_i]_A$;
- $[x_{D+i}]_A := (-2m_{D+i} \hat{\times} [p_i]_A) \hat{+} [k_{D+i} - m_{D+i}]_A$;
- $\left[x'_i\right]_A := (2m'_i \hat{\times} [p_i]_A) \hat{+} \left[k'_i\right]_A$;
- $\left[x'_{D+i}\right]_A := (-2m'_{D+i} \hat{\times} [p_i]_A) \hat{+} \left[k'_{D+i}\right]_A$;
- $[y_i]_A := (2m_i \hat{\times} [q_i]_A) \hat{+} [k_i]_A$;
- $[y_{D+i}]_A := (-2m_{D+i} \hat{\times} [q_i]_A) \hat{+} [k_{D+i}]_A$;
- $\left[y'_i\right]_A := (2m'_i \hat{\times} [q_i]_A) \hat{+} \left[k'_i + m'_i\right]_A$;
- $\left[y'_{D+i}\right]_A := (-2m'_{D+i} \hat{\times} [q_i]_A) \hat{+} \left[k'_{D+i} - m'_{D+i}\right]_A$;

Since the Paillier cryptosystem cannot decrypt negative values directly, we consider that each $k_{D+i_{\in K}}$, $k'_{D+i_{\in K'}}$, $m_{D+i_{\in M}}$, and $m'_{D+i_{\in M'}}$, must satisfy the conditions ($k_{D+i} > 2m_{D+i} \times Max_i$) and ($k'_{D+i} > 2m'_{D+i} \times Max_i$) for ($i = 1, \cdots, D$), during their generation process. Here Max_i indicates the maximum estimated i^{th} dimension attribute value of the objects. After expansion, the dominance relation between two encrypted objects $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$ will be turned to two vector comparison problems: (1) compare vector $[\boldsymbol{X}]_A$ and $[\boldsymbol{Y}]_A$, and (2) compare vector $\left[\boldsymbol{X'}\right]_A$ and $\left[\boldsymbol{Y'}\right]_A$.

Party$_B$ also creates two $2D$ length binary vectors $\boldsymbol{V}$ and $\boldsymbol{V'}$ to mark the expected comparison result between $[\boldsymbol{X}]_A$ and $[\boldsymbol{Y}]_A$, and between $\left[\boldsymbol{X'}\right]_A$ and $\left[\boldsymbol{Y'}\right]_A$. Particularly, $v_i = 1$

indicates $\boldsymbol{Party}_B$'s expectation of $x_i > y_i$ in position i, and $v_i = 0$ indicates $\boldsymbol{Party}_B$'s expectation of $x_i < y_i$. On the other hand, $v'_i = 1$ represents $\boldsymbol{Party}_B$'s expectation of $x'_i < y'_i$ in position i, whereas $v'_i = 0$ represents $\boldsymbol{Party}_B$'s expectation of $x'_i > y'_i$.

Next, $\boldsymbol{Party}_B$ generates two $2D$ length random binary vector $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma'}$ to swap the vector elements randomly and also to compute $\boldsymbol{W}$ and $\boldsymbol{W'}$ according to Step 3 of Algorithm 2. Considering two 2-dimensional encrypted object $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$ of Table 9, Table 10 presents the computation results of Step 1 to Step 3 of Algorithm 2.

TABLE 9. Encrypted object $E_A(P)$ and $E_A(Q)$.

object	d_1	d_2
$E_A(\boldsymbol{P})$	$[6]_A$	$[4]_A$
$E_A(\boldsymbol{Q})$	$[6]_A$	$[7]_A$

TABLE 10. Example of Step 1 to Step 3 of Algorithm 2.

	Dimension					Dimension			
vector	1	2	3	4	vector	1	2	3	4
$\boldsymbol{M}$	4	9	6	10	$\boldsymbol{M'}$	6	9	8	15
$\boldsymbol{K}$	26	17	170	218	$\boldsymbol{K'}$	22	12	211	289
$[\boldsymbol{X}]_A$	$[78]_A$	$[98]_A$	$[92]_A$	$[128]_A$	$[\boldsymbol{X'}]_A$	$[94]_A$	$[93]_A$	$[115]_A$	$[169]_A$
$[\boldsymbol{Y}]_A$	$[74]_A$	$[143]_A$	$[98]_A$	$[78]_A$	$[\boldsymbol{Y'}]_A$	$[100]_A$	$[147]_A$	$[107]_A$	$[64]_A$
$\boldsymbol{V}$	1	1	0	0	$\boldsymbol{V'}$	1	1	0	0
$\boldsymbol{\sigma}$	1	0	1	1	$\boldsymbol{\sigma'}$	0	1	1	0
$[\boldsymbol{X}]_A$	$[74]_A$	$[98]_A$	$[98]_A$	$[78]_A$	$[\boldsymbol{X'}]_A$	$[94]_A$	$[147]_A$	$[107]_A$	$[169]_A$
$[\boldsymbol{Y}]_A$	$[78]_A$	$[143]_A$	$[92]_A$	$[128]_A$	$[\boldsymbol{Y'}]_A$	$[100]_A$	$[93]_A$	$[115]_A$	$[64]_A$
$\boldsymbol{W}$	0	1	1	1	$\boldsymbol{W'}$	1	0	1	0

After that, to enhance the security through the data perturbation, $\boldsymbol{Party}_B$ generates four D length vectors of nonzero random integer: $\boldsymbol{\alpha}_{\in\mathbb{Z}^{>0}}$, $\boldsymbol{\beta}_{\in\mathbb{Z}^{>0}}$, $\boldsymbol{\alpha'}_{\in\mathbb{Z}^{>0}}$, and $\boldsymbol{\beta'}_{\in\mathbb{Z}^{>0}}$, s.t., $\alpha_{i\in\boldsymbol{\alpha}} \neq \beta_{i\in\boldsymbol{\beta}}$ and $\alpha'_{i\in\boldsymbol{\alpha'}} \neq \beta'_{i\in\boldsymbol{\beta'}}$. $\boldsymbol{Party}_B$ also creates two binary vector: $\boldsymbol{\rho}$ and $\boldsymbol{\rho'}$, and set $\rho_{i_{\in\rho}}$ and $\rho'_{i_{\in\rho'}}$ according to Step 4 of Algorithm 2.

Then, $\boldsymbol{Party}_B$ concatenates $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha'}$, $\boldsymbol{\beta'}$, and $\boldsymbol{\rho'}$ with $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{W}$, $\boldsymbol{X'}$, $\boldsymbol{Y'}$, and $\boldsymbol{W'}$, respectively. $\boldsymbol{Party}_B$ also generates random permutation function π and π' to shuffle the elements of concatenated vectors to obtain $[S]_A$, $[\boldsymbol{T}]_A$, $\boldsymbol{G}$, $[\boldsymbol{S'}]_A$, $[\boldsymbol{T'}]_A$, and $\boldsymbol{G'}$ according to Step 5 of Algorithm 2. After shuffling the vectors, $\boldsymbol{Party}_B$ uses a hash function to compute the hash values h and h' of binary vectors $\boldsymbol{G}$ and $\boldsymbol{G'}$, and sends $[S]_A$, $[\boldsymbol{T}]_A$, h, $[\boldsymbol{S'}]_A$, $[\boldsymbol{T'}]_A$, and h' to $\boldsymbol{Party}_A$.

Following Table 10 and assuming $\boldsymbol{\alpha} = (148, 165)$, $\boldsymbol{\beta} = (172, 140)$, $\boldsymbol{\alpha'} = (118, 154)$, and $\boldsymbol{\beta'} = (103, 136)$, Table 11 and Table 12 present the computation results of Step 4 to Step 6 of Algorithm 2. Here, we consider $\pi = (4, 6, 2, 5, 3, 1)$ and $\pi' = (3, 5, 1, 6, 2, 4)$. According to the equation $[S]_A \leftarrow \pi\left(\left[X|\boldsymbol{\alpha}\right]_A\right)$ of Step 5 and considering the first element of π (*i.e.* 4) implies that after shuffling, the first vector element of $[S]_A$ will be the fourth element of concatenated vector $\left[X|\boldsymbol{\alpha}\right]_A$.

TABLE 11. Example of Step 4 to Step 6 of Algorithm 2 (Part 1).

	Dimension					
vector	1	2	3	4	5	6
$[\boldsymbol{X}\vert\boldsymbol{\alpha}]_A$	$[74]_A$	$[98]_A$	$[98]_A$	$[78]_A$	$[148]_A$	$[165]_A$
$[\boldsymbol{Y}\vert\boldsymbol{\beta}]_A$	$[78]_A$	$[143]_A$	$[92]_A$	$[128]_A$	$[172]_A$	$[140]_A$
$\boldsymbol{W}\vert\boldsymbol{\rho}$	0	1	1	1	0	1
π	4	6	2	5	3	1
$[\boldsymbol{S}]_A$	$[78]_A$	$[165]_A$	$[98]_A$	$[148]_A$	$[98]_A$	$[74]_A$
$[\boldsymbol{T}]_A$	$[128]_A$	$[140]_A$	$[143]_A$	$[172]_A$	$[92]_A$	$[78]_A$
$\boldsymbol{G}$	1	1	1	0	1	0
h	$H\,(111010)$					

TABLE 12. Example of Step 4 to Step 6 of Algorithm 2 (Part 2).

	Dimension					
vector	1	2	3	4	5	6
$[\boldsymbol{X'}\vert\boldsymbol{\alpha'}]_A$	$[94]_A$	$[147]_A$	$[107]_A$	$[169]_A$	$[118]_A$	$[154]_A$
$[\boldsymbol{Y'}\vert\boldsymbol{\beta'}]_A$	$[100]_A$	$[93]_A$	$[115]_A$	$[64]_A$	$[103]_A$	$[136]_A$
$\boldsymbol{W'}\vert\boldsymbol{\rho'}$	0	1	1	1	0	1
π'	3	5	1	6	2	4
$[\boldsymbol{S'}]_A$	$[107]_A$	$[118]_A$	$[94]_A$	$[154]_A$	$[147]_A$	$[169]_A$
$[\boldsymbol{T'}]_A$	$[115]_A$	$[103]_A$	$[100]_A$	$[136]_A$	$[93]_A$	$[64]_A$
$\boldsymbol{G'}$	1	0	1	0	0	0
h'	$H\,(101000)$					

After receiving the encrypted vectors along with the expected hash values, $\boldsymbol{Party}_A$ decrypts the vectors using the key sk_A and obtains the plaintexts of $\boldsymbol{S}$, $\boldsymbol{T}$, $\boldsymbol{S'}$ and $\boldsymbol{T'}$. Although $\boldsymbol{Party}_A$ can compare the elements of the decrypted vectors, it will be quite impossible for $\boldsymbol{Party}_A$ to reproduce the original objects due to the anonymization of the vector elements through arbitrary transformation and data perturbation.

From the decrypted vectors, $\boldsymbol{Party}_A$ constructs the binary vectors $\boldsymbol{U}$ and $\boldsymbol{U'}$ according to Step 9 of Algorithm 2. Then, by comparing $H(\boldsymbol{U})$ with h, and $H(\boldsymbol{U'})$ with h', $\boldsymbol{Party}_A$ computes the dominance relation between two vectors $\boldsymbol{S}$ and $\boldsymbol{T}$ according to Step 10 to Step 12 of Algorithm 2. Table 13 shows the construction of $\boldsymbol{U}$ and $\boldsymbol{U'}$ from the decrypted vectors for our running example.

TABLE 13. Example of Step 9 of Algorithm 2.

	Dimension							Dimension					
vector	1	2	3	4	5	6	vector	1	2	3	4	5	6
$\boldsymbol{S}$	78	165	98	148	98	74	$\boldsymbol{S'}$	107	118	94	154	147	169
$\boldsymbol{T}$	128	140	143	172	92	78	$\boldsymbol{T'}$	115	103	100	136	93	64
$\boldsymbol{U}$	0	1	0	0	1	0	$\boldsymbol{U'}$	1	0	1	0	0	0

After comparing h with $H(\boldsymbol{U})$, and h' with $H(\boldsymbol{U'})$, $\boldsymbol{Party}_A$ can find $h \neq H(\boldsymbol{U})$ but $h' = H(\boldsymbol{U'})$, which implies that $\boldsymbol{S} \prec \boldsymbol{T}$. Therefore, $\boldsymbol{Party}_A$ sets $dom_S := 0$ and $dom_T := 1$ according to Step 11 of Algorithm 2, which ultimately reflects the dominance comparison result between two objects $\boldsymbol{P}$ and $\boldsymbol{Q}$. By examining the actual attributes of two encrypted objects $\boldsymbol{P}$ and $\boldsymbol{Q}$ within Table 9, we can also affirm $\boldsymbol{P} \prec \boldsymbol{Q}$.

Algorithm 3 Multi-party Skyline (MPS) protocol

```
Input: Each party has its local skyline objects, key pair, and public encryption keys of other parties;
    • $\boldsymbol{Party}_A$ has $Sky\,(\boldsymbol{DS}_A)$, $(pk_A, sk_A)$, and $pk_B, pk_C, \cdots$;
    • $\boldsymbol{Party}_B$ has $Sky\,(\boldsymbol{DS}_B)$, $(pk_B, sk_B)$, and $pk_A, pk_C, \cdots$;
    • $\boldsymbol{Party}_C$ has $Sky\,(\boldsymbol{DS}_C)$, $(pk_C, sk_C)$, and $pk_A, pk_B, \cdots$;
    ...
Output: Each party identifies its global skyline objects;
1:  Each party obtains its number of dominant objects in encrypted form for each local skyline object of other parties through
    the DOC protocol;
    • $\boldsymbol{Party}_A$ obtains $\left[\boldsymbol{dcount}_B^A\right]_B$, $\left[\boldsymbol{dcount}_C^A\right]_C, \cdots$;
    • $\boldsymbol{Party}_B$ obtains $\left[\boldsymbol{dcount}_A^B\right]_A$, $\left[\boldsymbol{dcount}_C^B\right]_C, \cdots$;
    • $\boldsymbol{Party}_C$ obtains $\left[\boldsymbol{dcount}_A^C\right]_A$, $\left[\boldsymbol{dcount}_B^C\right]_B, \cdots$;
    ...
    For $Sky\,(\boldsymbol{DS}_A)$ of $\boldsymbol{Party}_A$:
2:  Each party generates random integer $\boldsymbol{r}_{\in\mathbb{Z}^{>1}}$. After that
    $\boldsymbol{Party}_B$ computes $\left[f_A^B\right]_A := \left[\boldsymbol{dcount}_A^B\right]_A \hat{\times} \boldsymbol{r}$;
    $\boldsymbol{Party}_C$ computes $\left[f_A^C\right]_A := \left[\boldsymbol{dcount}_A^C\right]_A \hat{\times} \boldsymbol{r}$;
    ...
3:  if Number of parties = 2 then        ▷ Only $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$ are computing multi-party skyline
4:      $\boldsymbol{Party}_B$ sends $\left[f_A^B\right]_A$ to $\boldsymbol{Party}_A$;
5:      $\boldsymbol{Party}_A$ decrypts $\left[f_A^B\right]_A$ using $sk_A$ and identifies $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ as a global skyline object if $f_{A,i}^B = 0$;
6:  else        ▷ More than two parties are computing multi-party skyline
7:      A party $\boldsymbol{Party}_Z$ $\left(\boldsymbol{Party}_Z \neq \boldsymbol{Party}_A\right)$ collects $\left[f_A^B\right]_A$ from $\boldsymbol{Party}_B$, $\left[f_A^C\right]_A$ from $\boldsymbol{Party}_C, \cdots$;
8:      $\boldsymbol{Party}_Z$ computes $\left[\sum f_A\right]_A := \left[f_A^B\right]_A \hat{+} \left[f_A^C\right]_A \hat{+} \cdots \hat{+} \left[f_A^Z\right]_A$;
9:      $\boldsymbol{Party}_Z$ sends $\left[\sum f_A\right]_A$ to $\boldsymbol{Party}_A$;
10:     $\boldsymbol{Party}_A$ decrypts $\left[\sum f_A\right]_A$ using $sk_A$ and identifies each $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ as the global skyline object, if $\sum f_{A,i} = 0$;
11: end if
```

In order to prevent $\boldsymbol{Party}_B$ to know the dominance relation between two objects, $\boldsymbol{Party}_A$ also encrypts dom_S and dom_T using pk_A before sending them to $\boldsymbol{Party}_B$. Finally, $\boldsymbol{Party}_B$ assigns $[dom_S]_A$ and $[dom_T]_A$ to $[dom_P]_A$ and $\left[dom_Q\right]_A$, respectively.

C. MULTI-PARTY SKYLINE (MPS) PROTOCOL

The MPS protocol computes the global skyline from the privacy-preserving multi-party datasets. Each party identifies its global skyline objects through the MPS protocol described in Algorithm 3. Here, we explain how a party, *e.g.*, $\boldsymbol{Party}_A$ can identify its own global skyline objects. In the same way, other parties can also identify their global skyline objects.

At first, each party computes its number of dominant objects in encrypted form for other parties' local skyline objects through the DOC protocol. After that, according to Step 2 of Algorithm 3, each party multiplies a random integer $\boldsymbol{r}_{\in\mathbb{Z}^{>1}}$ with the encrypted dominant objects counter value obtained for each local skyline objects of other parties. Thus any party is unable to know precisely how many objects of other parties dominate its dominated local skyline objects. To explain our proposed framework, we denote this encrypted value as the masked dominant objects counter. Based on the number of participating parties, we design the rest of the MPS protocol as follows:

When the number of parties is two: If two parties, *i.e.*, $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$ are involved in the computation, then $\boldsymbol{Party}_B$ sends the encrypted value of masked dominant objects counter $\left[f_A^B\right]_A$ to $\boldsymbol{Party}_A$. After receiving $\left[f_A^B\right]_A$ from $\boldsymbol{Party}_B$, $\boldsymbol{Party}_A$ decrypts $\left[f_A^B\right]_A$ using the key sk_A and identifies each $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ as a global skyline object if $f_{A,i}^B = 0$. Considering column $\left[\boldsymbol{dcount}_A^B\right]_A$ of Table 7, Table 14 represents the

TABLE 14. Example of the MPS protocol for two parties.

	$\boldsymbol{Party}_B$			$\boldsymbol{Party}_A$
id	$[\boldsymbol{dcount}^B_A]_A$	$\boldsymbol{r}$	$[\boldsymbol{f}^B_A]_A$	$\boldsymbol{f}^B_A$
$\boldsymbol{A}_1$	$[0]_A$	8	$[0]_A$	**0**
$\boldsymbol{A}_2$	$[2]_A$	6	$[12]_A$	12
$\boldsymbol{A}_3$	$[1]_A$	4	$[4]_A$	4
$\boldsymbol{A}_4$	$[0]_A$	10	$[0]_A$	**0**

computation results from which $\boldsymbol{Party}_A$ identifies its global skyline objects.

When the number of parties is more than two: We consider that one of the participating parties acts as the coordinator in this scenario. The primary responsibility of this coordinator is to select a collector who collects the encrypted value of the masked dominant objects counters for one party's local skyline objects from other parties. The coordinator must not select the owner of the local skyline objects as the collector of the encrypted masked dominant objects counters of those local skyline objects.

Suppose, the coordinator selects $\boldsymbol{Party}_Z$ (one of the parties other than $\boldsymbol{Party}_A$) as the collector of the encrypted value of the masked dominant objects counters for the local skyline objects of $\boldsymbol{Party}_A$. Therefore, the other parties send the encrypted value of the masked dominant objects counters for the local skyline objects of $\boldsymbol{Party}_A$ to $\boldsymbol{Party}_Z$. Then, $\boldsymbol{Party}_Z$ computes the encrypted sum of masked dominant objects counters (*i.e.*, $\left[\sum \boldsymbol{f}_A\right]_A$) according to Step 8 of Algorithm 3, and sends it to $\boldsymbol{Party}_A$. After receiving, $\boldsymbol{Party}_A$ decrypts $\left[\sum \boldsymbol{f}_A\right]_A$, and identifies each $\boldsymbol{A}_{i \in Sky(\boldsymbol{DS}_A)}$ as the global skyline object **if** $\sum f_{A,i} = 0$.

Let us consider, four parties ($\boldsymbol{Party}_A$, $\boldsymbol{Party}_B$, $\boldsymbol{Party}_C$, and $\boldsymbol{Party}_D$) want to identify their global skyline objects. Let $\boldsymbol{Party}_A$ has three local skyline objects: $\boldsymbol{A}_1, \boldsymbol{A}_2$, and $\boldsymbol{A}_3$. Among these three objects, one object of $\boldsymbol{Party}_B$ and two objects of $\boldsymbol{Party}_C$ dominate $\boldsymbol{A}_1$; one object of $\boldsymbol{Party}_D$ dominates $\boldsymbol{A}_2$; none of the object of other parties dominates $\boldsymbol{A}_3$. Further assume, the coordinator selects $\boldsymbol{Party}_D$ as the collector of the encrypted values of the masked dominant objects counters for the local skyline objects of $\boldsymbol{Party}_A$. Based on these, Fig. 5 shows a data-flow diagram of the MPS protocol. Besides, Table 15 describes the computation results for the local skyline objects of $\boldsymbol{Party}_A$.

VI. PRIVACY AND SECURITY ANALYSES

In this section, we analyze the privacy and security aspects of the proposed framework. According to the composition theorem [29], a framework is considered as secure as long as its elemental protocols are secure, alongside all the intermediate results are random or pseudo-random. Now, we analyze the underlying protocols of our proposed framework.

A. PRIVACY OF THE DOC PROTOCOL

According to Algorithm 1, $\boldsymbol{Party}_B$ randomly shuffles the list of object pairs before comparing the dominance relation. Thus, $\boldsymbol{Party}_A$ cannot know which of its local skyline objects is being compared through the SDC protocol. Moreover, $\boldsymbol{Party}_B$ randomizes the parameters' sequence of the SDC protocol during dominance comparison. Therefore, by decrypting the anonymized data within the SDC protocol, $\boldsymbol{Party}_A$ cannot know whither $\boldsymbol{Party}_A$'s object dominates $\boldsymbol{Party}_B$'s object or vice versa.

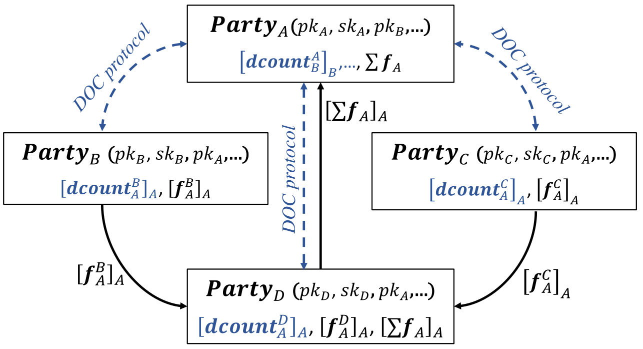
$\boldsymbol{Party}_A(pk_A, sk_A, pk_B,\ldots)$
$[\boldsymbol{dcount}^A_B]_B,\ldots, \sum \boldsymbol{f}_A$
DOC protocol
DOC protocol
$[\sum \boldsymbol{f}_A]_A$
$\boldsymbol{Party}_B\ (pk_B, sk_B, pk_A,\ldots)$
$[\boldsymbol{dcount}^B_A]_A, [\boldsymbol{f}^B_A]_A$
DOC protocol
$\boldsymbol{Party}_C\ (pk_C, sk_C, pk_A,\ldots)$
$[\boldsymbol{dcount}^C_A]_A, [\boldsymbol{f}^C_A]_A$
$[\boldsymbol{f}^B_A]_A$
$\boldsymbol{Party}_D\ (pk_D, sk_D, pk_A,\ldots)$
$[\boldsymbol{dcount}^D_A]_A, [\boldsymbol{f}^D_A]_A, [\sum \boldsymbol{f}_A]_A$
$[\boldsymbol{f}^C_A]_A$

FIGURE 5. Data-flow diagram of the MPS protocol for more than two parties.

On the other hand, $\boldsymbol{Party}_A$ encrypts the dominance comparison result of the SDC protocol before sending it to $\boldsymbol{Party}_B$. Consequently, $\boldsymbol{Party}_B$ cannot know the dominance relation between two specific objects. Besides, $\boldsymbol{Party}_B$ adds a nonzero random integer $\boldsymbol{r}$ with each $\left[\boldsymbol{dcount}^A_B\right]_A$. As a result, by decrypting $\left[\boldsymbol{e}^A_B\right]_A$, $\boldsymbol{Party}_A$ cannot know anything about the local skyline objects of $\boldsymbol{Party}_B$.

B. PRIVACY OF THE SDC PROTOCOL

As stated in Algorithm 2, $\boldsymbol{Party}_B$ generates four arrays of random integers $\boldsymbol{M}$, $\boldsymbol{K}$, $\boldsymbol{M'}$, and $\boldsymbol{K'}$ to construct vectors $\boldsymbol{X}$, $\boldsymbol{X'}$, $\boldsymbol{Y}$, and $\boldsymbol{Y'}$. After that, $\boldsymbol{Party}_B$ swaps the vector elements based on the random binary vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma'}$. Besides, $\boldsymbol{Party}_B$ also concatenates random integer vectors with the constructed vectors, and then shuffles it using random permutation function π and π'.

Since $\boldsymbol{Party}_A$ does not know which specific object of $\boldsymbol{Party}_A$ is being compared via the SDC protocol; without knowing $\boldsymbol{M}$, $\boldsymbol{R}$, $\boldsymbol{M'}$, $\boldsymbol{R'}$, $\boldsymbol{\sigma}$, $\boldsymbol{\sigma'}$, π, and π', $\boldsymbol{Party}_A$ cannot retrieve the object of $\boldsymbol{Party}_B$ only from the decrypted vectors. On the other hand, $\boldsymbol{Party}_A$ encrypts the dominance comparison result before sending it to $\boldsymbol{Party}_B$. Thereby, $\boldsymbol{Party}_B$ cannot know the dominance relation between two specific objects. Thus, the SDC protocol can ensure required data privacy for both parties while they compare the dominance relation between their objects.

C. PRIVACY OF THE MPS PROTOCOL

According to Algorithm 3, every party masks each of the encrypted dominant objects counters of other parties' objects by multiplying a random integer. Thus, all parties are unable to know precisely how many objects dominate each of their dominated objects.

Furthermore, when more than two parties compute the multi-party skyline, any party does not send the encrypted

TABLE 15. Example of the MPS protocol considering four parties.

	$Party_B$			$Party_C$			$Party_D$				$Party_A$
id	$[dcount^B_A]_A$	r	$[f^B_A]_A$	$[dcount^C_A]_A$	r	$[f^C_A]_A$	$[dcount^D_A]_A$	r	$[f^D_A]_A$	$[\sum f_A]_A$	$\sum f_A$
A_1	$[1]_A$	12	$[12]_A$	$[2]_A$	6	$[12]_A$	$[0]_A$	8	$[0]_A$	$[24]_A$	24
A_2	$[0]_A$	4	$[0]_A$	$[0]_A$	10	$[0]_A$	$[1]_A$	18	$[18]_A$	$[18]_A$	18
A_3	$[0]_A$	15	$[0]_A$	$[0]_A$	18	$[0]_A$	$[0]_A$	9	$[0]_A$	$[0]_A$	**0**

TABLE 16. Notations used for complexity analyses.

Notation	Definition
T_E	Complexity of homomorphic encryption
T_D	Complexity of homomorphic decryption
T_+	Complexity of homomorphic addition
$T_\times$	Complexity of homomorphic multiplication
T_-	Complexity of homomorphic subtraction
T_π	Complexity of vector permutation function
T_H	Complexity of hashing function
M	Number of parties for multi-party skyline query
N_A	Number of $Party_A$'s local skyline objects
N_B	Number of $Party_B$'s local skyline objects
B_H	Size of encrypted data
$B_\#$	Size of hash data

value of the masked dominant objects counters to the corresponding local skyline objects' owner individually. Therefore, any party cannot identify which and how many parties' object(s) dominates its specific local skyline object.

D. SECURITY OF THE PROPOSED FRAMEWORK

The proposed framework also maintains the security of the datasets of all participating parties. Within Fig. 3, Fig. 4, and Fig. 5, we can observe that all the exchanged data are being encrypted before transmission between the parties. Therefore, even if an adversary or an intruder eavesdrops on the communication media to obtain the transmitted data, it cannot get anything from the encrypted content.

VII. PERFORMANCE EVALUATION

Here, we analyze the complexity and evaluate the performance of the proposed framework. Also, we present a comparison of the proposed framework.

A. COMPLEXITY ANALYSES

To present the complexity analyses, we summarize the required notations in Table 16. Now the analyses of computation and communication complexity of DOC, SDC, and MPS protocols are presented in Table 17, Table 18, and Table 19, respectively.

B. EXPERIMENT

To evaluate the performance through simulation, we use two identical computers connected through Cisco Catalyst 2960-X Series Gigabit Switch, where one is considered as *Party*$_A$ and another as *Party*$_B$. Each computer is configured with an Intel® Core i5-6500 3.20GHz CPU, 8GB memory, and 64-bit Ubuntu 16.04 operating system. We develop our

TABLE 17. Complexity of the DOC protocol (based on Algorithm 1).

Step #	Complexity
Step 1	$N_A \cdot D \cdot (T_E + B_H)$
Step 2	$N_B \cdot D \cdot T_E$
Step 3	T_E
Step 5 - 9	$N_A \cdot N_B \cdot (T^*_{SDC} + 2T_+)$
Step 10	$N_B \cdot (T_+ + 2 \cdot T_E)$
Step 11	$2 \cdot N_B \cdot B_H$
Step 12 - 13	$N_B \cdot (T_D + T_E + T_-)$

* T_{SDC}: Total complexity of SDC protocol (Table 18)

TABLE 18. Complexity of the SDC protocol (based on Algorithm 2).

Step #	Complexity
Step 1	$8 \cdot D \cdot (T_\times + T_E + T_+)$
Step 4	$4D \cdot T_E$
Step 5	$6T_\pi$
Step 6	$2T_H$
Step 7	$12D \cdot B_H + 2B_\#$
Step 8	$12D \cdot T_D$
Step 10 - 13	$2T_H$
Step 14	$2B_H$

TABLE 19. Complexity of the MPS protocol (based on Algorithm 3).

Step #	Complexity
Step 1	Total complexity of computation through the DOC protocol
Global skyline objects identification by $Party_A$	
Step 2	$(M-1) \cdot N_A \cdot T_\times$
	For two parties (Step 4-5)
Step 4	$N_A \cdot B_H$
Step 5	$N_A \cdot T_D$
	For more than two parties (Step 7-10)
Step 7	$(M-2) \cdot N_A \cdot B_H$
Step 8	$(M-2) \cdot N_A \cdot T_+$
Step 9	$N_A \cdot B_H$
Step 10	$N_A \cdot T_D$

program using Java Remote Method Invocation (RMI) framework and use 80-bit Paillier encryption key. We generate synthetic datasets for our experiment where each attribute value of the synthetic datasets is randomly picked from 32-bit unsigned integer.

Initially, we extract two sets of local skyline objects from the generated datasets to represent the local skyline objects of two parties. After that, we examine the effect of dominance comparison through the SDC protocol within the DOC protocol. Since the number of dominance comparison within the DOC protocol depends on the number of two parties' local skyline objects, we vary the number of both parties' local skyline objects during our experiment. We also vary the object dimension from 2 to 5. Based on these, Fig. 6 shows the runtime of the DOC protocol. From the figure, it is seen that the runtime is linearly proportional to the number of dominance comparison through the SDC protocol as well as the number of object dimension, which is apparent since the complexity of the SDC protocol depends on the number of object dimension. Although every party can compute with all other parties through the DOC protocol within the MPS protocol, it does not require to maintain any specific synchronization. Hence, we do not evaluate the runtime of the MPS protocol.

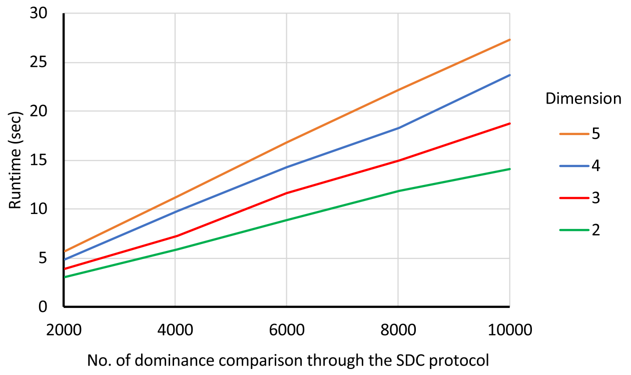

FIGURE 6. Runtime of the DOC protocol.

C. COMPARISONS

The proposed framework utilizes data anonymization and randomization schemes for secure dominance comparison. However, it does not lose the universality of the objects dominance relation. Thus, the utility of data and the skyline query results are not limited by the proposed framework. Also, many multi-party computation systems include one or more trusted third parties. It is a severe risk to the system if the third party(s) has been compromised. Whereas, the proposed framework does not utilize such a trusted third party. Furthermore, every party firstly computes the local skyline objects set from its database in plaintext space. Therefore, it significantly reduces the complexity of multi-party computation.

So far as we know, only one framework [7] computes privacy-preserving multi-party skyline without incorporating any semi-honest trusted third party. For this reason, we compare with this one. The ESVC protocol proposed in [7] depends on the length of the attribute value in the number of binary bits since it adapts the 0-encoding and 1-encoding scheme for comparing two integer vector elements. In contrast, we substituted the secure integer comparison with data anonymization schemes within our SDC protocol. Thereby, the complexity of our SDC protocol does not depend on the attribute value length. Also, the ESVC protocol requires five rounds of data exchanges, whereas our SDC protocol requires only two rounds of data exchanges during secure dominance comparison. Thus, our SDC protocol is more efficient than that of the ESVC protocol. To compare the performance, we simulate both protocols for the dominance comparison of the two-dimensional dataset objects. Fig. 7 shows the runtime comparison of the proposed SDC protocol with the ESVC protocol. From the figure, we can see that the runtime of the ESVC protocol is much higher than the SDC protocol.

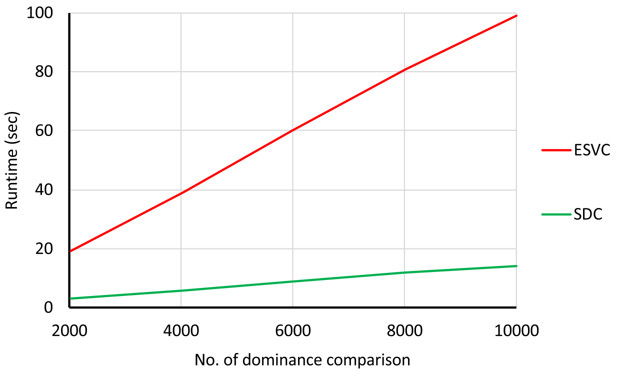

FIGURE 7. Runtime comparison of the proposed SDC protocol with the ESVC protocol [7]. Object dimension: 2, Attribute value length: 32-bit.

Moreover, the ESVC protocol discloses the dominance relation between two specific objects to both parties. Whereas, our SDC protocol does not reveal the dominance relation to anyone. Thus, our proposed framework enriches data privacy.

VIII. CONCLUSION

In this paper, we propose a novel framework for the skyline query considering the data privacy issues of multi-party data analyses. The detailed explanation of the proposed framework, along with the proper examples of the underlying protocols, confirms that all participating organizations can recognize their multi-party skyline objects without disclosing their dataset objects to others. The privacy and security analyses demonstrate that the framework satisfies the desired privacy requirements. Also, through extensive performance evaluation, we show the efficiency of the proposed framework for real-world deployment. Due to the avoidance of 'secure integer comparison', and the exploitation of encryption of 'the dominance comparison result' within our SDC protocol, our proposed framework becomes significantly efficient and secure. A future plan of improvement is to extend the work for computing other variants of skyline query, *e.g.*, k-skyband query (returns the set of objects that are dominated by at most k other objects), top k-dominating queries (returns the set of top-k objects that dominate the maximum number of objects), in a privacy-preserving way.

REFERENCES

[1] S. Borzsony, D. Kossmann, and K. Stocker, "The Skyline operator," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2001, pp. 421–430.

[2] X. Han, J. Li, D. Yang, and J. Wang, "Efficient skyline computation on big data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2521–2535, Nov. 2013.

[3] A. Zaman, M. A. Siddique, A. Annisa, and Y. Morimoto, "Secure computation of skyline query in *MapReduce*," in *Advanced Data Mining and Applications*, J. Li, X. Li, S. Wang, J. Li, and Q. Z. Sheng, Eds. Cham, Switzerland: Springer, 2016, pp. 345–360.

[4] W. Chen, M. Liu, R. Zhang, Y. Zhang, and S. Liu, "Secure outsourced skyline query processing via untrusted cloud service providers," in *Proc. IEEE INFOCOM 35th Annu. Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[5] J. Liu, J. Yang, L. Xiong, and J. Pei, "Secure skyline queries on cloud platform," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 633–644.

[6] J. Hua, H. Zhu, F. Wang, X. Liu, R. Lu, H. Li, and Y. Zhang, "CINEMA: Efficient and privacy-preserving online medical primary diagnosis with skyline query," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1450–1461, Apr. 2018.

[7] X. Liu, R. Lu, J. Ma, L. Chen, and H. Bao, "Efficient and privacy-preserving skyline computation framework across domains," *Future Gener. Comput. Syst.*, vol. 62, pp. 161–174, Sep. 2016, doi: 10.1016/j.future.2015.10.005.

[8] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," in *Proc. Proc. Int. Conf. Very Large Data Bases (VLDB)*, 2002, pp. 275–286.

[9] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, Mar. 2003, pp. 717–719.

[10] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 41–82, 2005.

[11] W.-T. Balke, U. Güntzer, and J. X. Zheng, "Efficient distributed skylining for Web information systems," in *Advances in Database Technology—EDBT*. Berlin, Germany: Springer, 2004, pp. 256–273.

[12] S. Wang, B. C. Ooi, A. K. H. Tung, and L. Xu, "Efficient skyline query processing on peer-to-peer networks," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 1126–1135.

[13] L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu, "iSky: Efficient and progressive skyline computing in a structured P2P network," in *Proc. 28th Int. Conf. Distrib. Comput. Syst.*, Jun. 2008, pp. 160–167.

[14] J. B. Rocha, A. Vlachou, C. Doulkeridis, and K. Nørvåg, "AGiDS: A grid-based strategy for distributed skyline query processing," in *Data Management in Grid and Peer-to-Peer Systems*. Berlin, Germany: Springer, 2009, pp. 12–23.

[15] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, 1982, pp. 160–164.

[16] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proc. 19th Annu. ACM Symp. Theory Comput. (STOC)*, 1987, pp. 218–229.

[17] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 2000, pp. 36–54.

[18] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 439–450.

[19] Z. Lin and J. W. Jaromczyk, "An efficient secure comparison protocol," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, Jun. 2012, pp. 30–35.

[20] T. Veugen, F. Blom, S. J. A. de Hoogh, and Z. Erkin, "Secure comparison protocols in the semi-honest model," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1217–1228, Oct. 2015.

[21] F. Kerschbaum, D. Biswas, and S. de Hoogh, "Performance comparison of secure comparison protocols," in *Proc. 20th Int. Workshop Database Expert Syst. Appl.*, Aug./Sep. 2009, pp. 133–136.

[22] H.-Y. Lin and W.-G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," in *Applied Cryptography and Network Security*, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Berlin, Germany: Springer, 2005, pp. 456–466.

[23] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, New York, NY, USA: ACM, 2018, pp. 1655–1658, doi: 10.1145/3183713.3197390.

[24] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 638–649.

[25] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Privacy aware learning," *J. ACM*, vol. 61, no. 6, pp. 38-1–38-57, Dec. 2014, doi: 10.1145/2666468.

[26] J. C. Duchi, M. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, Oct. 2013, pp. 429–438.

[27] K. Hose and A. Vlachou, "A survey of skyline processing in highly distributed environments," *Int. J. Very Large Data Bases*, vol. 21, no. 3, pp. 359–384, Jun. 2012, doi: 10.1007/s00778-011-0246-6.

[28] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Adv. Cryptol.-Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, J. Stern, Ed. Berlin, Germany: Springer, 1999, pp. 223–238.

[29] G. Oded, *Foundations of Cryptography: Basic Applications*, vol. 2, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2009.

MAHBOOB QAOSAR received the B.Sc. and M.Sc. degrees from the University of Rajshahi, Bangladesh, in 2006 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Graduate School of Engineering, Hiroshima University, Japan. He joined as a Faculty Member with the Computer Science and Engineering Department, University of Rajshahi, in 2010. His research interests include big data security, privacy-preserving information retrieval, and applied cryptography.

KAZI MD. ROKIBUL ALAM received the B.Sc. degree in CSE from Khulna University, Bangladesh, in 1999, the M.Sc. degree in CSE from the Bangladesh University of Engineering and Technology (BUET), in 2004, and the Dr.Eng. degree in system design engineering from the University of Fukui, Japan, in 2010. He is currently a Visiting Researcher with Hiroshima University, Japan. He is also a Professor with the Department of Computer Science and Engineering (CSE), Khulna University of Engineering and Technology (KUET). His research interests include applied cryptography, information security, and machine learning.

ASIF ZAMAN received the B.Sc. degree (Hons.) and the M.Sc. degree from the Computer Science and Engineering Department, University of Rajshahi, Bangladesh, and the Ph.D. degree from Hiroshima University, Japan, in 2017. Since 2006, he has been serving as a Faculty Member with the Department of Computer Science and Engineering, University of Rajshahi, where he is currently an Associate Professor. His field of research interests include big-data engineering and secure mining of interesting data, as well as computer security.

CHEN LI received the B.S. degree in computer engineering from the Qingdao College, Ocean University of China, in 2012, and the M.S. degree in computer engineering and the Ph.D. degree from Hiroshima University, Japan, in 2016. He is currently a Visiting Researcher with Hiroshima University. Since 2014, he has been involved in performance evaluation and analysis. His research interests include performance evaluation, data mining, skyline query, and deep learning.

SALEH AHMED received the B.Sc. and M.Sc. degrees in computer science and engineering from the University of Rajshahi, Bangladesh, in 2005 and 2006, respectively, and the Ph.D. degree from Hiroshima University, Japan, in 2019. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh. His research interests include secure data mining, semi-order preserving encryption, and signal processing.

MD. ANISUZZAMAN SIDDIQUE received the B.Sc. and M.Sc. degrees from the University of Rajshahi, Bangladesh, in 2000 and 2002, respectively, and the D.Eng. degree from Hiroshima University, Japan, in 2010. Since 2002, he has been with the University of Rajshahi. He also completed two years of Postdoctoral Research work at Hiroshima University, from 2013 to 2015. He is currently an Associate Professor with the University of Rajshahi. His current research interests include skyline evaluation, privacy-preserving information retrieval, data mining, machine learning, and deep learning.

YASUHIKO MORIMOTO received the B.E., M.E., and Ph.D. degrees from Hiroshima University, in 1989, 1991, and 2002, respectively. From 1991 to 2002, he was with the IBM Tokyo Research Laboratory, where he worked for a data mining project and a multimedia database project. Since 2002, he has been with Hiroshima University, Japan, where he is currently a Professor. His current research interests include data mining, machine learning, geographic information system, and privacy-preserving information retrieval.