Study on Privacy-preserving Multi-party Computation of Skyline and its Variants

（スカイライン問合せ及びその関連問合せに関する個人情報保護に
配慮したマルチパーティ計算手法の研究）

by

Mahboob Qaosar

A dissertation submitted

Graduate School of Engineering, Hiroshima University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Information Engineering

under supervision of

Yasuhiko Morimoto

Department of Information Engineering

Graduate School of Engineering

Hiroshima University, Japan

March, 2020

When you want something,

all the universe conspires

in helping you to achieve it.

Paulo Coelho

Dissertation Summary

Big data is no longer considered merely a large amount of data. Instead, it is regarded as business-driven data with analysis capabilities due to long-term business value. Big data analyses are considered as a revolution in the field of Information Technology. Various organizations all over the world are utilizing advanced analysis techniques to gain new insights from their big data.

Selecting the most influential data objects or samples from a large database is the initial task of big data analyses. The analysis results can be of relatively little value if the samples are not representative of the population from which the results are determined. The skyline query and its variants are functions to find such representative objects.

In principle, the skyline query and its variants select the representative objects from a multi-dimensional database based on the dominance relation. The skyline query returns the non-dominated objects from a multi-dimensional database. Similarly, the K-skyband query returns those objects from a database that are not dominated by more than K objects from the given database. On the other hand, the top-k dominating query returns the k data objects from a multi-dimensional database which dominate the highest number of data objects in the given database.

Nowadays, multiple organizations dealing with similar kinds of services want to perform data analysis operations on the union of their databases, referred to as multi-party computation. The multi-party computation of the skyline and its variants can also provide benefits to the participating organization to recognize their most influential data objects. Since the database of individual parties may contain sensitive information relevant to their services or customers or products, any organization does not want to disclose its private database to others. However, it is not possible to compare the dominance relation between

multi-party database objects without revealing the database. Realizing this issue, the data-privacy is widely studied in this dissertation for the computation of the multi-party skyline and variants.

Various settings and approaches for the computation of the privacy-preserving multi-party skyline and variants are considered within this dissertation. Several relevant works are reviewed to achieve the goal. The computation and communication complexities of one of proposed frameworks is also analyzed in this study.

This dissertation begins with the discussion and background of the problem in **Chapter 1**. Then, some basic preliminaries and literature surveys on related topics of the dissertation are presented in **Chapter 2**. The rest of this dissertation is split into several parts.

At first, **Chapter 3** discusses a framework for the privacy-preserving multi-party skyline query. The significant advantage of the proposed framework is that it does not require any trusted third-party for the multi-party skyline query. The framework utilizes the Paillier cryptosystem along with the data anonymization and perturbation techniques for the multi-party computation. It also secures the intermediate computation results during the multi-party skyline query to ensure the highest privacy and security of the participating parties' databases.

After that, this dissertation proposes an efficient approach for the K-skyband query in distributed multi-party databases without unveiling the objects' attributes directly. This approach considers that all parties securely transform their objects' attributes without changing their sorting order rank on each dimension of the database objects. Then, a trusted third party computes the multi-party K-skyband from the transformed values of the objects' attributes. The detail of this process is explained in **Chapter 4**.

On the other hand, the top-k dominating query has drawn massive attention in the database community since it combines the advantages of the top-k query and the skyline query. The multi-party top-k dominating query can provide more benefits to the participating organizations to identify their most influential products or services. However, in the conventional computation system, it is not possible to compute the multi-party top-k dominating objects without revealing the individual parties' databases to others. Therefore, a framework for the cloud-based privacy-preserving multi-party top-k dominating query has been proposed in **Chapter 5**.

Finally, a concluding discussion with the future guideline to extend this research work is discussed in **Chapter 6**.

Acknowledgments

First of all, I would like to express my deep gratitude to the Almighty for bestowing His blessings and for enabling me to accomplish my research work successfully.

I want to express my sincere gratitude to my supervisor Prof. Yasuhiko Morimoto, Graduate School of Engineering, Hiroshima University, Japan for giving me the opportunity to explore new ideas in the field of secure data mining and privacy-preserving information retrieval. This research would not have been possible for me without his moral support and excellent guidance. His superior knowledge, motivation, and diligence expertise in this field provide me numerous opportunities to discover new things to develop my carrier as a researcher. Besides my supervisor, I would also like to express my appreciation to the rest members of my dissertation committee: Prof. Toru Nakanishi and Prof. Koji Eguchi for their valuable comments, encouragement, and critical judgment, which enriched my knowledge.

I am grateful to the Government of Japan for providing the scholarship for pursuing my research. Special thanks to all my co-researchers and lab members for their assistance and contributions during my research.

Last but not least, I am deeply grateful to my family members for their comprehension, dedication, sacrifices, and support throughout the research work.

Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

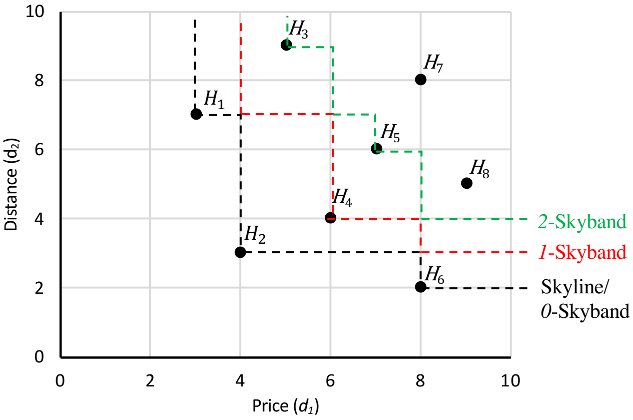
Data is playing a massive role in today's world. Therefore, data is considered as a commodity whose value is incalculable. Realizing these, different organizations throughout the world are producing, analyzing, and storing a vast amount of data, known as 'big data'. As a result, the demands for big data analysis tools are increasing significantly. These tools have attracted massive attention to the organizations and the researchers for making strategic decisions and for new knowledge discovery. Still, big data is introducing new challenges for collection, storage, process, analyze, etc.

With the rapidly growing data volume, many applications facing the problem of choosing the most influential data objects from these vast databases. In some cases, an objective ranking function can be used to sort the data objects by their relevance, *e.g.*, the top-10 objects showed by a web search engine. On the other hand, several applications consider more diverse preferences and multiple criteria to find good objects. Such applications can benefit from the computation of the skyline and its variants, which select the representative objects from a large multi-dimensional database based on the dominance relation.

In general, an object is said to be non-dominated if it is not worse than any other object in every attribute of the objects and is better in at least one attribute. Based on this,

ID	Price (d_1)	Distance (d_2)
H_1	3	7
H_2	4	3
H_3	5	9
H_4	6	4
H_5	7	6
H_6	8	2
H_7	8	8
H_8	9	5

(a)



(b)

Figure 1.1: Example of the dominance relation, the skyline query and its variants

the skyline query returns all non-dominated objects from a given database [4, 19], which is different from the traditional SQL queries returning a complete result set. Whereas, the K-skyband query returns those objects from a multi-dimensional database, which are dominated by at most K other objects within the database [32, 10]. On the other hand, the top-k dominating query, another variant of the skyline query, returns the k data objects from a database based on the rank of 'domination score' [18, 22], where the 'domination score' of an object is defined as how many objects in the database are dominated by the object.

Consider the example in Figure 1.1. It describes a table and a plot diagram of some resorts/hotels with their price and distance from the beach. By comparing the hotel $\boldsymbol{H}_4$ with another hotel $\boldsymbol{H}_5$, it can be observable that the hotel $\boldsymbol{H}_4$ is better than $\boldsymbol{H}_5$ in every dimension. In such a case, it is said that $\boldsymbol{H}_5$ is dominated by $\boldsymbol{H}_4$, or $\boldsymbol{H}_4$ is dominating $\boldsymbol{H}_5$. However, it is also understandable that $\boldsymbol{H}_4$ is also dominated by $\boldsymbol{H}_2$. The mathematical definitions of dominance relation, the skyline query, the k-skyband query, and the top-k dominating query are discussed in Chapter 2.

In this example, the skyline query retrieves $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, and $\boldsymbol{H}_6$ as the skyline (black line) since any other hotel in the table does not dominate them. Furthermore, Figure 1.1(b) also illustrates a 1-skyband query (red line) and a 2-skyband query (green line) of the hotels with a special case of $K = 0$ representing the original skyline. In detail, a 2-skyband query returns the skyline (which are not dominated any other hotel), $\boldsymbol{H}_4$ (which is dominated by at most one hotel), and $\boldsymbol{H}_3$, $\boldsymbol{H}_5$ (which are dominated by at most two hotels). Besides, since $\boldsymbol{H}_1$ dominates $\boldsymbol{H}_3$ and $\boldsymbol{H}_7$, it can be said that the 'domination score' of $\boldsymbol{H}_1$ is 2. Based on the 'domination score', the top-2 dominating query retrieves $\boldsymbol{H}_2$ and $\boldsymbol{H}_4$, since the 'domination score' of $\boldsymbol{H}_2$ is 5, the 'domination score' of $\boldsymbol{H}_4$ is 3, and the domination score of every other hotel is less than 3.

In the present century, E-commerce is growing at an unprecedented rate all over the world. Because of this, the competition among multiple organizations, who work with similar products and services, is also increasing significantly. To survive in this competitive market, they are successively adding innovative features and values to their products and services. They have also noticed the importance of query/analyzing results obtained from the union of databases owned by multiple organizations. Such joint query operations are often referred to as the *multi-party computation (MPC).*

Since the database of individual organizations or parties may contain sensitive and confidential information about their products/services/customers, any organization does not want to reveal its private database to others. However, they are willingly want to receive the results of query operations performed on their combined databases. In such a case, the organizations want to maintain the privacy of their individual database during the multi-party computation.

Like other multi-party computation problem, the multi-party computation of the skyline and its variants can help the participating parties/organizations by retrieving the results from the combined database of the participating parties/organizations. Such results of the multi-party computation can help the organizations to locate their most competitive products or services. However, such multi-party query operations also demand discloser of parties' private databases to others during the comparison of dominance relation between the different parties' objects. Therefore, in conventional computation, it is not possible to compute the multi-party skyline and its variants without revealing the objects in one party to others. Realizing the above problem of data privacy and security, this study proposes and analyzed various approaches for the privacy-preserving multi-party computation of skyline and its variants.

1.1 Motivation

In various data mining applications, different organizations or agents may want to recognize which of their objects are not only skyline objects of their individual database but also members of skyline objects of their combined database. However, the organizations do not want to reveal their database to others during the computation. Even they do not want to disclose how many of its database objects are in the multi-party skyline. Although some of the existing frameworks may resolve it, they involve one or more trusted third parties or the curator to solve their problem. Since even the third party(s) may involve in the conspiracy, it is difficult to imagine the unbiased third-party(s) who will be trusted by all parties. Therefore, a new framework for the privacy-preserving multi-party skyline query has been proposed in this research, which does not involve such kind of third parties. Through the proposed framework, although every party can identify its multi-party skyline objects without exposing the database to others, any party cannot know how many multi-party skyline objects own by other parties.

Although the skyline query returns a set of non dominated objects, many practical applications also demand to obtain the closest objects to the skyline objects from the database along with the skyline objects. In such a case, the K-skyband query can play an important role, which retrieves the objects from a multi-dimensional database that are dominated by at most K other objects within the database. Like other multi-party computation problems, the multi-party K-skyband query can also benefit the participating parties by retrieving the K-skyband objects from the union of multi-party databases. Since the computation of skyline and its variants depends on the dominance comparison between the database objects, very few works had been proposed for privacy-preserving skyline query by comparing the encrypted database objects securely on the cloud platform [26, 29, 16]. It is also apparent that depending on the value of K, the secure dominance comparison protocol based privacy-preserving skyline query will take a longer time for the K-skyband query. However, it can be possible to compute K-skyband by using the sorting order rank of the multi-dimensional objects' attributes on each dimension [7]. Therefore, an efficient framework has been proposed in this study to transform objects' attributes without changing their sorting order rank on each dimension of the object in a privacy-preserving multi-party computation environment. Then the framework utilizes the transformed values of the objects' attributes for the privacy-preserving multi-party K-skyband query.

Same as the skyline and the K-skyband query, the popularity of the top-k query is increasing day by day to extract a limited number of preferable objects from a large database. In general, the top-k query retrieves the k data objects that have better scores than others based on user-defined monotone scoring function. However, it is not easy for users to specify an appropriate scoring function. On the other hand, the top-k dominating query combines the advantages of the top-k query and the skyline query, eliminating their disadvantages. It is noticeable that the identification of the top-k dominating objects from the union of multi-party databases can undoubtedly benefit the participating parties. Since the privacy of the individual parties databases is a major barrier for such a multi-party top-k dominating query, this dissertation proposes another framework for the multi-party top-k dominating query in the privacy-preserving multi-party databases. This framework also ensures the privacy of the query results. Therefore, only the individual participating parties can recognize their qualified multi-party top-k dominating objects.

1.2 Thesis Organization

The rest of this dissertation is organized as follows: **Chapter 2** discusses the mathematical definitions and properties of the skyline, K-skyband, and top-k dominating queries as well as the Paillier cryptosystem. This chapter also introduces some basic notations which are frequently used throughout this dissertation and reviews some related works on skyline query, privacy-preserving multi-party computation, and privacy-preserving multi-party skyline query. Then **Chapter 3** presents a framework for privacy-preserving multi-party skyline query which does not require any third-party for the multi-party computation. After that, an efficient framework for privacy-preserving K-skyband query in distributed multi-party databases is discussed in **Chapter 4**. Here the author introduces a novel technique for transforming the multi-party objects' attributes without altering their sorting order rank and utilizes the transformed values of the objects' attributes for the multi-party K-skyband query. Then **Chapter 5** introduces another framework for the privacy-preserving multi-party top-k dominating query, which not only preserves the privacy of the database objects but also keeps the privacy of the query results. At last, **Chapter 6** concludes this dissertation. The possible future direction to extend this research work is also discussed here.

Chapter 2

Background Studies

This chapter discusses some preliminaries which are required to understand this dissertation and also reviews of some related works.

2.1 Preliminaries

This section explains the mathematical definitions and properties of the skyline query and its variants. It also discusses the Paillier cryptosystem and introduces some basic notations which are frequently used throughout this dissertation article.

2.1.1 Definitions and Properties of Skyline Query and its Variants

For the last few decades, the skyline query and its variants are recognized as the most popular and useful query tool in the database researchers community. These queries retrieve a set of representative objects from a large dataset based on the dominance relation. To define dominance relationship and skyline query and its variants mathematically, let us consider a dataset $\boldsymbol{DS}$ with D-dimensions $\{d_1, d_2, \cdots, d_D\}$. This dissertation uses $\boldsymbol{O}_i.d_j$ to denote the j-th dimension value of object $\boldsymbol{O}_i$. Without losing generality, it is assumed that the smaller value in each attribute is better.

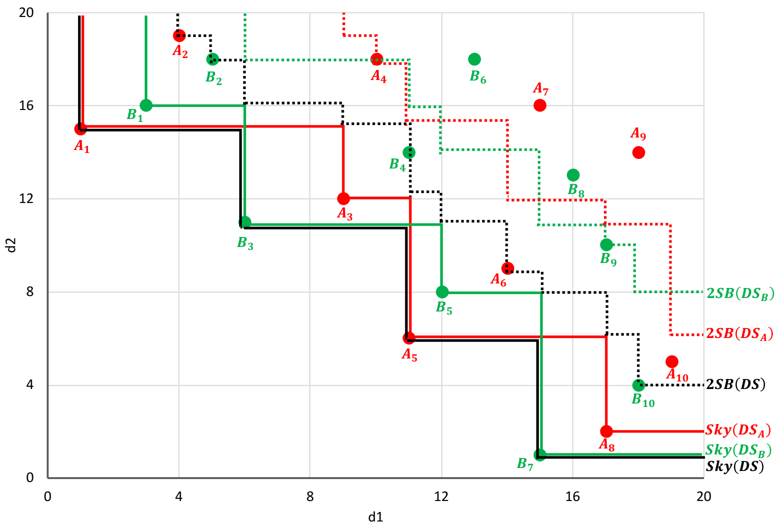


Figure 2.1: The skyline query and its variants with their properties

- **Dominance:** An object $\boldsymbol{O}_i \in \boldsymbol{DS}$ is said to dominate another object $\boldsymbol{O}_j \in \boldsymbol{DS}$, denoted as $\boldsymbol{O}_i \prec \boldsymbol{O}_j$, if $\boldsymbol{O}_i.d_r \leq \boldsymbol{O}_j.d_r$ $(1 \leq r \leq D)$ for all D attributes and $\boldsymbol{O}_i.d_t < \boldsymbol{O}_j.d_t$ $(1 \leq t \leq D)$ for at least one attribute. In such a case, between $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$, $\boldsymbol{O}_i$ is called a dominant object, and $\boldsymbol{O}_j$ is called a dominated object. For the example given in Figure 2.1, $\boldsymbol{A}_1$ is better than $\boldsymbol{B}_1$ in every dimension. Therefore, $\boldsymbol{A}_1 \prec \boldsymbol{B}_1$.

- **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 $\boldsymbol{DS}$ plotted in Figure 2.1, objects $\boldsymbol{A}_1, \boldsymbol{B}_3, \boldsymbol{A}_5, \boldsymbol{B}_7$ are not dominated by any other objects. Therefore, the skyline query retrieves $Sky(\boldsymbol{DS}) = \{\boldsymbol{A}_1, \boldsymbol{B}_3, \boldsymbol{A}_5, \boldsymbol{B}_7\}$.

- **$\boldsymbol{K}$-Skyband:** An object $\boldsymbol{O}_i \in \boldsymbol{DS}$ is said to be a K-skyband object of $\boldsymbol{DS}$, if $\boldsymbol{O}_i$ is dominated by at most K objects in $\boldsymbol{DS}$. This dissertation describes the K-skyband objects set of $\boldsymbol{DS}$ by $KSB(\boldsymbol{DS})$. In Figure 2.1, $\boldsymbol{B}_2$ is dominated only by $\boldsymbol{A}_1$ and $\boldsymbol{B}_1$ in $\boldsymbol{DS}$. Thus, $\boldsymbol{B}_2 \in 2SB(\boldsymbol{DS})$. The black dotted line in Figure 2.1 depicts the boundary of $2SB(\boldsymbol{DS})$.

- **$\boldsymbol{\mu}$ Score/Domination Score:** The μ score or the domination score of an object $\boldsymbol{O}_i$ describes how many objects in the dataset are dominated by $\boldsymbol{O}_i$. The the μ score of an object $\boldsymbol{O}_i$ is denoted by $\mu(\boldsymbol{O}_i)$. In Figure 2.1, object $\boldsymbol{A}_1$ dominates six objects $(\boldsymbol{B}_1, \boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{A}_4, \boldsymbol{B}_6, \boldsymbol{A}_7)$. Therefore, the μ score of $\boldsymbol{A}_1$ is 6, *i.e.*, $\mu(\boldsymbol{A}_1) = 6$.

- **Top-$\boldsymbol{k}$ Dominating Query:** Given a positive integer k and a dataset $\boldsymbol{DS}$, the top-k dominating query returns the k objects from $\boldsymbol{DS}$ that have the highest μ scores. For the dataset in Figure 2.1, $\boldsymbol{A}_5$ dominates eight objects, and both $\boldsymbol{B}_3$ and $\boldsymbol{B}_7$ dominates seven objects. The rest of the other objects dominates less than seven objects. Therefore, the top-3 dominating query retrieves $\boldsymbol{A}_5$, $\boldsymbol{B}_3$ and $\boldsymbol{B}_7$.

This dissertation also utilizes some important properties of skyline, K-skyband, and top-k dominating queries to optimize the computation.

- **Additivity of the Skyline Query[15]:**

Given a dataset $\boldsymbol{DS}$ and n datasets $\boldsymbol{DS}_i (i = 1, \cdots, n)$ such that $\boldsymbol{DS} = \bigcup_{i=1}^{n} \boldsymbol{DS}_i$, the following equation holds:

$$Sky(\boldsymbol{DS}) = Sky\left(\bigcup_{i=1}^{n} Sky(\boldsymbol{DS}_i)\right)$$

This implies that each skyline object of $\boldsymbol{DS}$ must be a skyline object of $\boldsymbol{DS}$'s subset.

Let us consider, the red and the green points in Figure 2.1 represent the objects of $\boldsymbol{DS}_A$ and $\boldsymbol{DS}_B$, respectively. The skyline objects of $\boldsymbol{DS}_A$ and $\boldsymbol{DS}_B$ is given as $Sky(\boldsymbol{DS}_A) = \{\boldsymbol{A}_1, \boldsymbol{A}_3, \boldsymbol{A}_5, \boldsymbol{A}_8\}$ and $Sky(\boldsymbol{DS}_B) = \{\boldsymbol{B}_1, \boldsymbol{B}_3, \boldsymbol{B}_5, \boldsymbol{B}_7\}$. It is apparent that the common skyline objects is given as $Sky(\boldsymbol{DS}) = Sky(\boldsymbol{DS}_A \cup \boldsymbol{DS}_B) = \{\boldsymbol{A}_1, \boldsymbol{B}_3, \boldsymbol{A}_5, \boldsymbol{B}_7\}$, where $\{\boldsymbol{A}_1, \boldsymbol{A}_5\} \in Sky(\boldsymbol{DS}_A)$ and $\{\boldsymbol{B}_3, \boldsymbol{B}_7\} \in Sky(\boldsymbol{DS}_B)$.

- **Additivity of the $\boldsymbol{K}$-Skyband Query:**

Given a dataset $\boldsymbol{DS}$ and n datasets $\boldsymbol{DS}_i (i = 1, \cdots, n)$ such that $\boldsymbol{DS} = \bigcup_{i=1}^{n} \boldsymbol{DS}_i$, the following equation holds:

$$KSB(\boldsymbol{DS}) = KSB\left(\bigcup_{i=1}^{n} KSB(\boldsymbol{DS}_i)\right)$$

Proof of the Additivity of $\boldsymbol{K}$-Skyband Query:

Assume that $\boldsymbol{DS}_1, \boldsymbol{DS}_2, \cdots, \boldsymbol{DS}_n$ be n disjoint datasets. We define the dataset $\boldsymbol{DS} = \bigcup\limits_{i=1}^{n} \boldsymbol{DS}_i$.

Let us consider an object $\boldsymbol{O} \in \boldsymbol{DS}^i (i = 1, 2, \cdots, n)$, where $\boldsymbol{DS}_i \subset \boldsymbol{DS}$. If $\boldsymbol{O}$ is dominated by more than K objects of $\boldsymbol{DS}_i$, then $\boldsymbol{O} \notin KSB(\boldsymbol{DS}_i)$. In such a case, it can be referred that $\boldsymbol{O}$ is also dominated by more than K objects of $\boldsymbol{DS}$, since all objects in $\boldsymbol{DS}_i$ also belong to combined dataset $\boldsymbol{DS}$. Therefore, an object that is not a K-skyband object of $\boldsymbol{DS}_i$ cannot be a K-skyband object of $\boldsymbol{DS}$.

From the above discussion, it is proved that every object of $KSB(\boldsymbol{DS})$ must be the member of $\bigcup\limits_{i=1}^{n} KSB(\boldsymbol{DS}_i)$. Furthermore, since the dominance relation is transitive, every non-K-skyband must be dominated by more than K number of K-skyband objects of the corresponding dataset. Therefore, it is sufficient to perform the K-skyband operation only on $\bigcup\limits_{i=1}^{n} KSB(\boldsymbol{DS}_i)$ to obtain $KSB(\boldsymbol{DS})$. It concludes $KSB(\boldsymbol{DS}) = KSB\left(\bigcup\limits_{i=1}^{n} KSB(\boldsymbol{DS}_i)\right)$.

In Figure 2.1, the 2-skyband objects of $\boldsymbol{DS}_A$ and $\boldsymbol{DS}_B$ can be given by $2SB(\boldsymbol{DS}_A) = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4, \boldsymbol{A}_5, \boldsymbol{A}_6, \boldsymbol{A}_8, \boldsymbol{A}_{10}\}$ and $2SB(\boldsymbol{DS}_B) = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3, \boldsymbol{B}_4, \boldsymbol{B}_5, \boldsymbol{B}_7, \boldsymbol{B}_9, \boldsymbol{B}_{10}\}$, whereas $2SB(\boldsymbol{DS}) = 2SB(2SB(\boldsymbol{DS}_A) \cup 2SB(\boldsymbol{DS}_B)) = \{\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{B}_3, \boldsymbol{A}_3, \boldsymbol{A}_5, \boldsymbol{B}_5, \boldsymbol{A}_6, \boldsymbol{B}_7, \boldsymbol{A}_8, \boldsymbol{B}_{10}\}$ can give the 2-skyband objects of dataset $\boldsymbol{DS} = \boldsymbol{DS}_A \cup \boldsymbol{DS}_B$.

The author also introduces and frequently uses two common terminologies throughout the dissertation: the *local skyline object*, the *global skyline object*, the *local K-skyband object*, and the *global K-skyband object*. Here the local skyline object denotes the skyline object of a sub-dataset, *i.e.*, an object of $Sky(\boldsymbol{DS}_A)$, while the global skyline object denotes the skyline object computed from the union of sub-datasets, *i.e.*, an object of $Sky(\boldsymbol{DS})$. Similarly, the local K-skyband object represents the K-skyband object of sub-dataset and the global K-skyband object represents the K-skyband object of union dataset.

- **A Property of Top-k Dominating Query:**

Top-k dominating data objects belong to $(k-1)$-skyband of the dataset.

Proof:

If an object $\boldsymbol{O}$ is dominated by more than or equal to k data objects in a dataset $\boldsymbol{DS}$, then it can be said that there exist at least k objects in $\boldsymbol{DS}$, whose μ-score is greater than $\mu(\boldsymbol{O})$. Therefore, any object which is dominated by more than or equal to k data objects cannot be a top-k dominating object. In other words, every top-k dominating object is dominated by at most $k-1$ objects in $\boldsymbol{DS}$.

According to the definition of K-skyband, every K-skyband object is dominated by at most K objects in $\boldsymbol{DS}$. It concludes, the top-k dominating data objects belong to $(k-1)$-skyband of the dataset.

2.1.2 Paillier Cryptosystem and its Properties

Paillier cryptosystem [31] is a probabilistic asymmetric algorithm for public key cryptography. The scheme is an additive homomorphic cryptosystem, *i.e.*, given only the public key and the encryption of two plaintext integer m_1 and m_2, one can compute the encryption of $m_1 + m_2$. This dissertation vastly utilizes the Paillier cryptosystem along with data anonymization schemes for the privacy-preserving multi-party computation.

In the Paillier cryptosystem, the public encryption key and the private decryption key is given as $pk(N, G)$ and $sk(\lambda, \mu)$, respectively. The key generation algorithm along with the encryption and decryption processes of the Paillier cryptosystem can be found in [31].

Additive Homomorphism Properties of Paillier Cryptosystem:

Assume, m_1 and m_2 be two distinct plaintext integers while $[m_1]$ and $[m_2]$ represent their ciphertext, respectively. Based on this, the additive homomorphism properties of Paillier cryptosystem can be given as follows:

- Homomorphic Addition

$$[m_1 + m_2] := ([m_1] \times [m_2]) \bmod N^2$$

- Homomorphic Multiplication

$$[c \times m_1] := ([m_1])^c \bmod N^2$$

- Homomorphic Subtraction

$$[m_1 - m_2] := \left([m_1] \times [m_2]^{N-1}\right) \bmod N^2$$

Here N is the part of Paillier public encryption key and c is a constant integer.

2.1.3 Useful Notations

Table 4.1 introduced some common notations used in this dissertation.

2.2 Related Works

The earlier studies of skyline query and its variants, privacy-preserving multi-party computation, and privacy-preserving skyline query instigate this dissertation. Following Subsection 2.2.1 reviews some of the proposed methods for processing skyline query and its variants. Then Subsection 2.2.2 introduces some studies on privacy-preserving multi-party computation, followed by Subsection 2.2.3 highlights some of the existing works on the privacy-preserving multi-party skyline query.

Notation	Definition
D	Object Dimension
$\boldsymbol{O}, \boldsymbol{A}, \boldsymbol{B}, \cdots$	Object/Vector/Array
$\boldsymbol{A} \prec \boldsymbol{B}$	Dominance Relation between $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{A}\vert\boldsymbol{B}$	Concatenation of Vectors $\boldsymbol{A}$ and $\boldsymbol{B}$
pk, sk	Public encryption key and Private decryption key
$[\boldsymbol{x}]$	Ciphertext of $\boldsymbol{x}$ (Encrypted by the Public key pk)
$\boldsymbol{Party}_P$	P^{th} Participating party
$\boldsymbol{DS}_P$	Dataset of $\boldsymbol{Party}_P$
$\vert\boldsymbol{DS}_P\vert$	Number of Objects in $\boldsymbol{DS}_P$
$Sky\left(\boldsymbol{DS}_P\right)$	Skyline Objects of $\boldsymbol{Party}_P$
$KSB\left(\boldsymbol{DS}_P\right)$	K-skyband Objects of $\boldsymbol{Party}_P$
pk_P, sk_P	Public encryption key and Private decryption key of $\boldsymbol{Party}_P$
$[\boldsymbol{x}]_P$	$\boldsymbol{x}$ is Encrypted by the Public key pk_P of $\boldsymbol{Party}_P$;
$\mathbb{Z}$	Universal set of Integers
$\hat{+}$, $\hat{-}$, $\hat{\times}$	Homomorphic Addition, Subtraction, Multiplication
$\pi(\cdot)$, $\pi'(\cdot)$, $\Pi(\cdot)$	Random Permutation Function

Table 2.1: Summary of Notations

2.2.1 Skyline, K-Skyband and Top-k Dominating Queries

Borzsony et al. first introduced the Skyline operator, which was used in a query to filter results from a database to select only those objects that are not worse than any other [4]. They proposed three algorithms to process skyline over large databases, which are *Block-Nested-Loops (BNL)*, *Divide-and-Conquer (D&C)*, and B-tree-based schemes. The *BNL* algorithm compares the dominance relation between every pair of objects within a database. If any other object in the given database does not dominate an object, the *BNL* algorithm lists that object as a skyline object. Whereas, by taking the problem of the memory limitation of the system into consideration, the *D&C* algorithm divides the large database into several partitions that can fit into the system memory. The skyline for each partition is then computed, and the final skyline is produced by applying skyline operation on the merged skyline results of each partition. Later, by improving the *D&C* algorithm, Kossmann et al. proposed the *nearest-neighbor (NN)* algorithm [19]. This algorithm prunes out the dominated objects efficiently by iteratively partitioning the data space based on the nearest objects in the space. On the other hand, Chomicki et al. improved BNL by presorting, known as Sort-Filter-Skyline (SFS) [7]. On the basis of the *Best First Nearest Neighbor (BF-NN)* algorithm, Papadias et al. proposed another progressive algorithm for the skyline query, known as the *Branch-and-Bound Skyline (BBS)* [32].

Similar to the skyline query, some works utilized the K-skyband query for data extraction. Gao et al. proposed an algorithm for the K-skyband query to extract representative objects from an incomplete database [10]. Whereas, to obtain the top-k query from a large database efficiently, Gong et al. apply the K-skyband query on the database objects for reducing the search space [13].

Several algorithms had also been proposed for top-k dominating query. Kontaki et al. proposed the top-k dominating query on the continuous streaming database [18]. Lian and Chen research on the processing of probabilistic top-k dominating query in the uncertain database [22]. To process the top-k dominating query efficiently, Yiu and Mamoulis introduced a batch counting technique for computing the domination score of multiple objects simultaneously and also proposed a priority-based tree traversal algorithm [43]. Similarly, Zhang et al. proposed an efficient, threshold-based algorithm to compute the top-k dominating query accurately [45].

Recently, the distributed computing paradigm is gaining popularity for the computation of the skyline and its variants. Balke et al. introduced several models for the distributed skyline query on the vertically partitioned web information [3]. Both Wang et al. and Chen et al. researched the skyline query in structured P2P networks, where the individual peers are responsible for a partial region of data space [6, 40]. Alternatively, Rocha-Junior et al. also proposed a grid-based approach for distributed skyline processing(AGiDS), where each peer maintains a grid-based data summary structure for describing its data distribution [34]. The MapReduce framework also had also been proposed for efficient skyline computation in some articles [20, 30, 35]. Similarly, by taking advantage of the distributed computing environment, several algorithms enhanced the efficiency of the top-k dominating query [2, 5, 9]. However, these works did not consider the privacy and security of distributed data objects.

2.2.2 Privacy-preserving Multi-party Computation

Privacy-preserving multi-party computation is a subfield of cryptography aiming to create protocols for the parties to jointly compute a function over their inputs while keeping those inputs private [41]. Unlike conventional cryptographic tasks, where cryptography assures security and integrity of communication or storage, the cryptography in this model preserves participants' privacy from each other. Yao was the first introducer of such kind of multi-party computation problem for the two-party setting [42]. After that, Goldreich et al. and many others expand this problem for more than two parties [12]. According to [12], privacy in multi-party computation means that the participants' input data remain secret throughout the secure function evaluation process, and the participants could only receive the computed results of the function. Basically, the privacy-preserving multi-party computation protocols are relatively complex compared to specific purpose protocols.

Privacy-preserving data mining is an essential aspect of various data mining applications. Therefore, it had been studied significantly to achieve some data mining goals without compromising the privacy of the individuals. Lindell and Pinkas proposed an algorithm for privacy-preserving data mining operation on the combined databases of two parties, where one party does not reveal its database to another [25]. They utilized secure multi-party computation protocols to solve the problem. Agrawal et al. defined a data mining problem in a different way, where one party wants to conduct a data-mining operation on a private database owned by another party [1]. They applied the data perturbation schemes to prevent the querying party from accessing precise information in individual data records of the data owner.

Several studies on the privacy-preserving multi-party computation utilized homomorphic encryption schemes for comparing the private data [24, 39], although they are highly expensive *w.r.t.* computation and communication complexity [17]. Besides, Lin et al. also introduced another secure comparison protocol known as the 0-encoding and 1-encoding scheme [23], which is a two-party secure comparison protocol for comparing two integers in two rounds of data exchange. Similar to [24] and [39], the complexity of this scheme also depends on the length of the integer attribute value in the number of binary bits.

Several studies had been proposed for privacy-preserving multi-party computation in the cloud platform where the participants outsource their encrypted databases and the query operations to a group of trusted cloud providers. The cloud providers assure the privacy of the encrypted database as well as the clients' queries. Considering such a cloud computation scenario, Elmehdwi et al. proposed a solution to the k-nearest neighbor (kNN) query problem over the outsourced encrypted database [8]. Whereas, Rahulamathavan et al. introduced a privacy-preserving multi-class support vector machine for outsourced encrypted data in the cloud [33]. Liu et al. introduced a privacy-preserving clinical decision support system based on the Naïve Bayesian (NB) classifier [27]. A privacy-preserving deep learning scheme had been introduced by Li et al. in [21]. The above works addressed the privacy of both the users' dynamic queries and the encrypted database during multi-party computation.

Besides, several database queries could be applied to the rank/sorting-order of the objects' attributes, *e.g.*, skyline query and its variants, querying with an aggregate function, statistical analysis, and so on [14, 38, 37, 44]. Hamada et al. proposed oblivious radix sort for ranking multi-party objects securely [14]. However, this scheme requires multiple computation and communication rounds between the coordinator and the data owner for ranking the objects' attributes without disclosing them to others. Recently Xin et al. also proposed a solution for the secure multi-party objects' attributes ranking problem. [28]. However, their algorithm assumes that the attributes' values belong to a universal set; thus, the complexity of their algorithm depends on the cardinality of the predefined universal set.

2.2.3 Secure Skyline Query

Because of the information security and privacy awareness, secure data analyses are becoming a key research issue in 'big data' processing. Similarly, the secure skyline query is also being studied for mining big data securely, realizing various application perspectives.

Addressing the privacy of the variant skyline queries of the users, three different frameworks were proposed by Liu et al. [26] and Hua et al. [16]. By using their proposed frameworks, the database owner will be unable to know the users' queries. On the other hand, the users also cannot know anything about the secured database other than the query results. To compare the dominance relation, [26] integrates secure integer comparison and secure bit-decomposition protocols proposed by Veugen et al. [39] and Samanthula et al. [36]. Whereas, [16] reduced the communication overhead of secure comparison protocol by using the 0-encoding and 1-encoding scheme proposed by Lin et al. [23].

Liu et al. proposed a privacy-preserving multi-party skyline query framework to compute skyline in distributed multi-party databases, where any party does not disclose its database to others [29]. They adopt the 0-encoding and 1-encoding scheme [23] and introduce the Lightweight Additive Homomorphic Public Key Encryption(LAHE) Scheme to improve the performance of secure dominance comparison. They also utilize the additivity property of the skyline query to reduce the number of dominance comparisons between multi-party objects.

On the other hand, Zaman et al. proposed a secure objects' ranking-based skyline query framework [44]. Integrating the oblivious radix sort [14], their framework first transforms the objects' attributes in their rank on each object dimension and then utilize the objects' rank on each dimension for the multi-party skyline query. However, these framework requires multiple rounds of computation as well as data transmission for ranking the objects' attributes securely.

Chapter 3

Privacy-preserving Multi-party Skyline Query

This chapter introduces a privacy-preserving multi-party skyline query framework that does not incorporates any trusted semi-honest third party for the multi-party skyline query. Since the third parties may involve in the conspiracy, it is challenging to assume an unbiased third-party(s) who will be trusted by all parties. Considering this threat, a novel framework for the privacy-preserving multi-party skyline is proposed here. It does not employ any trusted third party for the multi-party computation. Therefore, it can overcome the circumstances where a dishonest party and a biased third party may collude with each other for revealing the private data of an honest party. Through this framework, only the data owner can identify whither its dataset object is a multi-party skyline object or not. Even, no party can know the number of multi-party skyline objects that other parties own.

The remaining part of this chapter is organized as follows: the proposed system model and the desired privacy requirement is specified in Section 3.1 and 3.2, respectively. Section 3.3 explain the detailed framework with necessary algorithms. Next, the privacy and security analysis for the proposed framework is discussed in Section 3.4. After that, Section 3.5 demonstrates the performance evaluation through complexity analyses, extensive simulations, and comprehensive comparison. Finally, Section 3.6 concludes this chapter.

3.1 System Model

During the system design stage, it is considered that all participating parties have sensitive datasets, and they are connected through data communication media. Without revealing the dataset to others, all participating parties want to identify their global skyline objects from their datasets that are not dominated by any object of their combined datasets. Maintaining the privacy of every participant's dataset during the multi-party skyline query is the primary concern for this system model. Here, the semi-honest adversary model is adopted, and it is assumed that all participants are honest-but-curious, *i.e.*, all participants strictly follow the protocol but intend to extract the sensitive data of other parties from the computation.

Due to the additivity property of the skyline query, it can be said that each object of the global skyline must be an object of any of the local skyline of the parties. Therefore, it is assumed 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 datasets, and thus improve the computation efficiency.

Assume, Table 3.1 represents the private datasets of three individual parties, while Table 3.2 shows their local skyline objects. After computing the local skyline, every party wants to identify its global skyline objects without revealing its local skyline objects to others. Based on Table 3.2, Table 3.3 derives the global skyline objects owned by individual parties.

Table 3.1: Local datasets of the individual parties

$\boldsymbol{DS}_A$ of $\boldsymbol{Party}_A$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{A_1}$	5	26
$\boldsymbol{A_2}$	10	16
$\boldsymbol{A_3}$	13	24
$\boldsymbol{A_4}$	16	11
$\boldsymbol{A_5}$	18	17
$\boldsymbol{A_6}$	25	15
$\boldsymbol{A_7}$	27	7

$\boldsymbol{DS}_B$ of $\boldsymbol{Party}_B$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{B_1}$	4	25
$\boldsymbol{B_2}$	10	20
$\boldsymbol{B_3}$	17	22
$\boldsymbol{B_4}$	20	13
$\boldsymbol{B_5}$	22	18
$\boldsymbol{B_6}$	25	5
$\boldsymbol{B_7}$	26	12

$\boldsymbol{DS}_C$ of $\boldsymbol{Party}_C$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{C_1}$	7	23
$\boldsymbol{C_2}$	11	27
$\boldsymbol{C_3}$	13	18
$\boldsymbol{C_4}$	16	25
$\boldsymbol{C_5}$	18	13
$\boldsymbol{C_6}$	21	22
$\boldsymbol{C_7}$	23	9

Table 3.2: Local skyline objects of the individual parties

$Sky\,(\boldsymbol{DS}_A)$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{A_1}$	5	26
$\boldsymbol{A_2}$	10	16
$\boldsymbol{A_4}$	16	11
$\boldsymbol{A_7}$	27	7

$Sky\,(\boldsymbol{DS}_B)$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{B_1}$	4	25
$\boldsymbol{B_2}$	10	20
$\boldsymbol{B_4}$	20	13
$\boldsymbol{B_6}$	25	5

$Sky\,(\boldsymbol{DS}_C)$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{C_1}$	7	23
$\boldsymbol{C_3}$	13	18
$\boldsymbol{C_5}$	18	13
$\boldsymbol{C_7}$	23	9

Table 3.3: Global skyline objects (GSO) of the individual parties

GSO of $\boldsymbol{Party}_A$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{A_2}$	10	16
$\boldsymbol{A_4}$	16	11

GSO of $\boldsymbol{Party}_B$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{B_1}$	4	25
$\boldsymbol{B_6}$	25	5

GSO of $\boldsymbol{Party}_C$		
id	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$
$\boldsymbol{C_1}$	7	23
$\boldsymbol{C_7}$	23	9

3.2 Desired Privacy

This 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 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 3.2, $\boldsymbol{Party}_A$ has no information about a global skyline object that is owned by $\boldsymbol{Party}_B$ or $\boldsymbol{Party}_C$. Also, $\boldsymbol{Party}_A$ cannot know how many global objects $\boldsymbol{Party}_B$ and $\boldsymbol{Party}_C$ 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 of others or not. According to Table 3.3, after secure computation, $\boldsymbol{Party}_A$ can identify that $\boldsymbol{A_2}$ is a global skyline object, but $\boldsymbol{Party}_A$ cannot know whether $\boldsymbol{A_2}$ 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 3.2 and Table 3.3, $\boldsymbol{Party}_B$ can determine $\boldsymbol{B_4}$ is not a global skyline object, but $\boldsymbol{Party}_B$ cannot know how many local skyline objects of $\boldsymbol{Party}_A$ or $\boldsymbol{Party}_C$ dominates $\boldsymbol{B_4}$.

- 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 $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$, $\boldsymbol{Party}_C$ can find that $\boldsymbol{C_3}$ is dominated by other parties' object(s). However, $\boldsymbol{Party}_C$ is unable to know: $\boldsymbol{C_3}$ is dominated by $\boldsymbol{Party}_A$'s object(s), or $\boldsymbol{Party}_B$'s object(s), or both parties' objects.

3.3 Proposed Framework

In the 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. Three intra-dependent protocols are designed to build the proposed framework. These are the Multi-Party Skyline (MPS) protocol, the Dominant Objects Counter (DOC) protocol, and the Secure Dominance Comparison (SDC) protocol. In this framework, the MPS protocol applies the DOC protocol among every pair of parties, while the DOC protocol utilizes the SDC protocol to securely compare the dominance relationships between the local skyline objects of two individual parties. The following subsections 3.3.1, 3.3.2, and 3.3.3, describe the DOC, the SDC, and the MPS protocols, respectively.

3.3.1 Dominant Objects Counter (DOC) Protocol

The DOC protocol is a two-party protocol. For every local skyline object of both parties, it securely counts the dominant objects within the opposite party's local skyline objects. Suppose, $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$ are two participating 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. Based on this scenario, Algorithm 1 briefly describes the DOC protocol and Fig. 3.1 depicts its data-flow diagram.

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 $\left[dc^B_{A,i}\right]_A$ and $\left[dc^A_{B,j}\right]_A$ 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 $dc^B_{A,i}$ counts the objects in $Sky\left(\boldsymbol{DS}_B\right)$, which dominates $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$. Similarly, $dc^A_{B,j}$ counts the objects in $Sky\left(\boldsymbol{DS}_A\right)$, which dominates $\boldsymbol{B}_{j\in Sky(\boldsymbol{DS}_B)}$.

Algorithm 1 Dominant Objects Counter (DOC) protocol

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

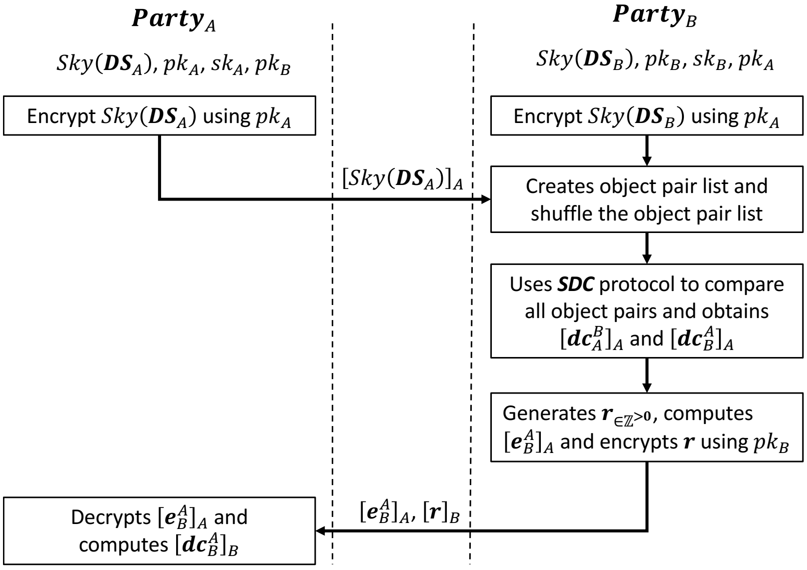

$\boldsymbol{Party}_A$
$Sky(\boldsymbol{DS}_A)$, pk_A, sk_A, pk_B
Encrypt $Sky(\boldsymbol{DS}_A)$ using pk_A
$\boldsymbol{Party}_B$
$Sky(\boldsymbol{DS}_B)$, pk_B, sk_B, pk_A
Encrypt $Sky(\boldsymbol{DS}_B)$ using pk_A
$[Sky(\boldsymbol{DS}_A)]_A$
Creates object pair list and shuffle the object pair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

Figure 3.1: Data-flow diagram of the DOC protocol

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.*, $Sky\left(\boldsymbol{DS}_A\right)\times Sky\left(\boldsymbol{DS}_B\right) = \{(\boldsymbol{A}_i, \boldsymbol{B}_j)\,|\boldsymbol{A}_i\in Sky\left(\boldsymbol{DS}_A\right)\ and\,\boldsymbol{B}_j\in Sky\left(\boldsymbol{DS}_B\right)\}$. 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 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, $\left[\boldsymbol{dc}_A^B\right]_A$ holds the number of dominant objects in $Sky\left(\boldsymbol{DS}_B\right)$ for each object of $Sky\left(\boldsymbol{DS}_A\right)$. Also, $\left[\boldsymbol{dc}_B^A\right]_A$ holds the number of dominant objects in $Sky\left(\boldsymbol{DS}_A\right)$ for each object of $Sky\left(\boldsymbol{DS}_B\right)$. 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 $\boldsymbol{dc}_A^B$, and $\boldsymbol{dc}_B^A$ are encrypted by pk_A. In contrast, $\boldsymbol{Party}_A$ also cannot determine the global skyline 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 object of $\boldsymbol{Party}_B$ in encrypted form. Therefore, $\boldsymbol{Party}_B$ first generates random positive integer array $\boldsymbol{r}_{\in\mathbb{Z}^+}$ and computes $\left[\boldsymbol{e}_B^A\right]_A := \left[\boldsymbol{dc}_B^A\right]_A \hat{+} [r]_A$ 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 $\boldsymbol{Party}_A$. 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 $\left[\boldsymbol{dc}_B^A\right]_B$ for $Sky\left(\boldsymbol{DS}_B\right)$ by computing through Step 12 and Step 13 of Algorithm 1.

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.

3.3.2 Secure Dominance Comparison (SDC) protocol

The SDC protocol is a sub-protocol of the DOC protocol. It is the principal component of the proposed framework, which is designed to compare the dominance relation between two parties' encrypted objects. Same as the DOC protocol, the SDC protocol is also explained considering two parties: $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$, where $\boldsymbol{Party}_A$ has the key pair (pk_A, sk_A),

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 $[dom_Q]_A$
        $\underline{\boldsymbol{Party}_B}$:
    Expands $[\boldsymbol{P}]_A$ and $[\boldsymbol{Q}]_A$ into four $2D$ length vector: $[\boldsymbol{X}]_A$, $[\boldsymbol{X'}]_A$, $[\boldsymbol{Y}]_A$ and $[\boldsymbol{Y'}]_A$;
    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 $\left[x_{i \in \boldsymbol{X}}\right]_A$ and $\left[y_{i \in \boldsymbol{Y}}\right]_A$ if $\sigma_i = 1$;
        ii. Swaps each element $\left[x'_{i \in \boldsymbol{X'}}\right]_A$ and $\left[y'_{i \in \boldsymbol{Y'}}\right]_A$ 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 $\rho_{i \in \boldsymbol{\rho}} = 1$ if $\alpha_{i \in \boldsymbol{\alpha}} > \beta_{i \in \boldsymbol{\beta}}$ else set $\rho_{i \in \boldsymbol{\rho}} = 0$;
        iii. Creates $D$ length binary vector $\boldsymbol{\rho'}$ and set $\rho'_{i \in \boldsymbol{\rho'}} = 1$ if $\alpha'_{i \in \boldsymbol{\alpha'}} < \beta'_{i \in \boldsymbol{\beta'}}$ else set $\rho'_{i \in \boldsymbol{\rho'}} = 0$;
        iv. Encrypts $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\alpha'}$, and $\boldsymbol{\beta'}$ using $pk_A$;
    i. Computes $[\boldsymbol{S}]_A \leftarrow \boldsymbol{\pi}\left([\boldsymbol{X}|\boldsymbol{\alpha}]_A\right)$, $[\boldsymbol{T}]_A \leftarrow \boldsymbol{\pi}\left([\boldsymbol{Y}|\boldsymbol{\beta}]_A\right)$, and $\boldsymbol{G} \leftarrow \boldsymbol{\pi}\left(\boldsymbol{W}|\boldsymbol{\rho}\right)$;
        ii. Computes $[\boldsymbol{S'}]_A \leftarrow \boldsymbol{\pi'}\left([\boldsymbol{X'}|\boldsymbol{\alpha'}]_A\right)$, $[\boldsymbol{T'}]_A \leftarrow \boldsymbol{\pi'}\left([\boldsymbol{Y'}|\boldsymbol{\beta'}]_A\right)$, and $\boldsymbol{G'} \leftarrow \boldsymbol{\pi'}\left(\boldsymbol{W'}|\boldsymbol{\rho'}\right)$;;
    Uses hash function to compute $h := H(\boldsymbol{G})$ and $h' := H(\boldsymbol{G'})$;
    Sends $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $h$, $[\boldsymbol{S'}]_A$, $[\boldsymbol{T'}]_A$, and $h'$ to $\boldsymbol{Party}_A$;
        $\underline{\boldsymbol{Party}_A}$:
    Decrypts $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $[\boldsymbol{S'}]_A$, and $[\boldsymbol{T'}]_A$ 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 $s_{i \in \boldsymbol{S}} > t_{i \in \boldsymbol{T}}$ then set $u_i := 1$ else set $u_i := 0$;
      ii. if $s'_{i \in \boldsymbol{S'}} < t'_{i \in \boldsymbol{T'}}$ 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 $\boldsymbol{Party}_B$;
        $\underline{\boldsymbol{Party}_B}$:
    Assigns $[dom_P]_A := [dom_S]_A$ and $[dom_Q]_A := [dom_T]_A$;
```

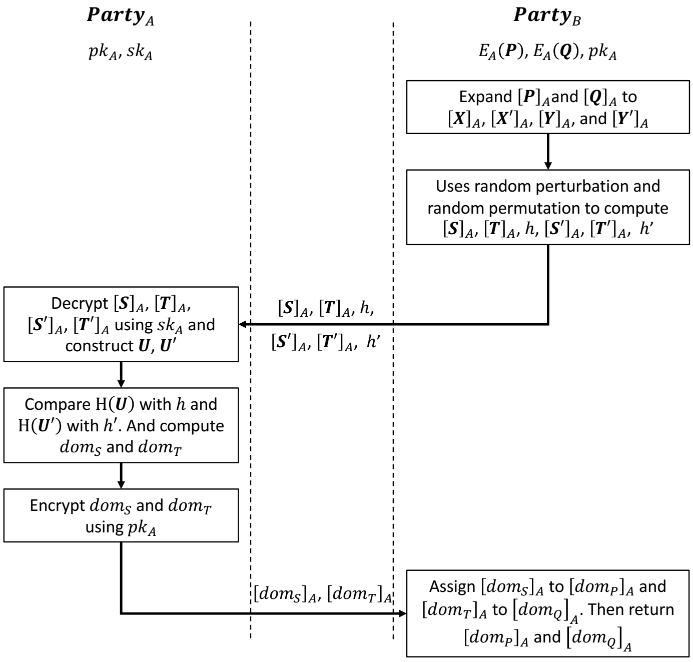

$\boldsymbol{Party}_A$
pk_A, sk_A
$\boldsymbol{Party}_B$
$E_A(\boldsymbol{P})$, $E_A(\boldsymbol{Q})$, pk_A
Expand $[\boldsymbol{P}]_A$and $[\boldsymbol{Q}]_A$ to $[\boldsymbol{X}]_A$, $[\boldsymbol{X}']_A$, $[\boldsymbol{Y}]_A$, and $[\boldsymbol{Y}']_A$
Uses random perturbation and random permutation 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 H($\boldsymbol{U}$) with h and 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 $[dom_Q]_A$. Then return $[dom_P]_A$ and $[dom_Q]_A$

Figure 3.2: Data-flow diagram of the SDC protocol

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.

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. The SDC protocol is designed obeying the basic principle of the ESVC Protocol [29]. However, to improve the computation efficiency, the 0-encoding and 1-encoding scheme based secure integer comparison protocol [23] is ignored here which was used in the ESVC protocol. Instead, the data anonymization, perturbation, and randomization techniques is adopted in the proposed SDC protocol. Furthermore, it is considered encrypting the dominance relation between two objects to maintain the desired privacy requirements described in Section 3.2. Algorithm 2 describes the SDC protocol and Fig. 3.2 depicts its data-flow diagram.

It is acknowledged that three types of dominance relationships between two objects $\boldsymbol{P}$

and $\boldsymbol{Q}$ are possible: 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 $[\boldsymbol{X}]_A$, $[\boldsymbol{X'}]_A$, $[\boldsymbol{Y}]_A$ and $[\boldsymbol{Y'}]_A$. In this regard, $\boldsymbol{Party}_B$ generates four $2D$ length random integer array to anonymize the vector elements using arbitrary transformation. These are $\boldsymbol{M} = (m_1, ..., m_{2D})_{\in \mathbb{Z}^{>1}}$, $\boldsymbol{M'} = (m'_1, ..., m'_{2D})_{\in \mathbb{Z}^{>1}}$, $\boldsymbol{K} = (k_1, ..., k_{2D})_{\in \mathbb{Z}^+}$, and $\boldsymbol{K'} = (k'_1, ..., k'_{2D})_{\in \mathbb{Z}^+}$.

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

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

Since the Paillier cryptosystem cannot decrypt negative values directly, it is considered that each $k_{D+i_{\in \boldsymbol{K}}}$, $k'_{D+i_{\in \boldsymbol{K'}}}$, $m_{D+i_{\in \boldsymbol{M}}}$, and $m'_{D+i_{\in \boldsymbol{M'}}}$ 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 $[\boldsymbol{X'}]_A$ and $[\boldsymbol{Y'}]_A$.

$\boldsymbol{Party_B}$ 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 $[\boldsymbol{X'}]_A$ and $[\boldsymbol{Y'}]_A$. 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.

After that, to enhance the security through the data perturbation, $\boldsymbol{Party_B}$ generates four D length vectors of nonzero random integer: $\boldsymbol{\alpha}_{\in\mathbb{Z}^+}$, $\boldsymbol{\beta}_{\in\mathbb{Z}^+}$, $\boldsymbol{\alpha'}_{\in\mathbb{Z}^+}$, and $\boldsymbol{\beta'}_{\in\mathbb{Z}^+}$, *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 vectors $\boldsymbol{\rho}$ and $\boldsymbol{\rho'}$, and set $\rho_{i_{\in\boldsymbol{\rho}}}$ and $\rho'_{i_{\in\boldsymbol{\rho'}}}$ 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 $\boldsymbol{\pi}$ and $\boldsymbol{\pi'}$ to shuffle the elements of concatenated vectors to obtain $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, $\boldsymbol{G}$, $[\boldsymbol{S'}]_A$, $[\boldsymbol{T'}]_A$, 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 $[\boldsymbol{S}]_A$, $[\boldsymbol{T}]_A$, h, $[\boldsymbol{S'}]_A$, $[\boldsymbol{T'}]_A$, and h' to $\boldsymbol{Party_A}$.

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 12 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 13 to

Step 16 of Algorithm 2. 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 $[dom_Q]_A$, respectively.

3.3.3 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, it is explained 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 this framework, this encrypted value is denoted as the masked dominant objects counter. Depending on the number of participating parties, the rest of the MPS protocol is designed 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 $\left[\boldsymbol{f}_A^B\right]_A$ to $\boldsymbol{Party}_A$. After receiving $\left[\boldsymbol{f}_A^B\right]_A$ from $\boldsymbol{Party}_B$, $\boldsymbol{Party}_A$ decrypts $\left[\boldsymbol{f}_A^B\right]_A$ using the key sk_A and identifies each $\boldsymbol{A}_{i \in Sky(\boldsymbol{DS}_A)}$ as a global skyline object if $f_{A,i}^B = 0$.

- **When the number of parties is more than two:** It is considered that one of the participating parties acts as the coordinator in this scenario. The primary responsibility

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;
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 $\left[\boldsymbol{dc}_B^A\right]_B$, $\left[\boldsymbol{dc}_C^A\right]_C$, $\cdots$;
    • $\boldsymbol{Party}_B$ obtains $\left[\boldsymbol{dc}_A^B\right]_A$, $\left[\boldsymbol{dc}_C^B\right]_C$, $\cdots$;
    • $\boldsymbol{Party}_C$ obtains $\left[\boldsymbol{dc}_A^C\right]_A$, $\left[\boldsymbol{dc}_B^C\right]_B$, $\cdots$;
    ...
    For $Sky\,(\boldsymbol{DS}_A)$ of $\boldsymbol{Party}_A$:
Each party generates random integer $\boldsymbol{r}_{\in\mathbb{Z}>1}$. After that
    $\boldsymbol{Party}_B$ computes $\left[\boldsymbol{f}_A^B\right]_A := \left[\boldsymbol{dc}_A^B\right]_A \hat{\times} \boldsymbol{r}$;
    $\boldsymbol{Party}_C$ computes $\left[\boldsymbol{f}_A^C\right]_A := \left[\boldsymbol{dc}_A^C\right]_A \hat{\times} \boldsymbol{r}$;
    ...
if Number of parties = 2 then        ▷ Only $\boldsymbol{Party}_A$ and $\boldsymbol{Party}_B$ are computing multi-party skyline
    $\boldsymbol{Party}_B$ sends $\left[\boldsymbol{f}_A^B\right]_A$ to $\boldsymbol{Party}_A$;
    $\boldsymbol{Party}_A$ decrypts $\left[\boldsymbol{f}_A^B\right]_A$ and identifies $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ as a global skyline object if $f_{A,i}^B = 0$;
else        ▷ More than two parties are computing multi-party skyline
    A party $\boldsymbol{Party}_Z$ ($\boldsymbol{Party}_Z \neq \boldsymbol{Party}_A$) collects $\left[\boldsymbol{f}_A^B\right]_A$ from $\boldsymbol{Party}_B$, $\left[\boldsymbol{f}_A^C\right]_A$ from $\boldsymbol{Party}_C$, $\cdots$;
    $\boldsymbol{Party}_Z$ computes $[\sum \boldsymbol{f}_A]_A := \left[\boldsymbol{f}_A^B\right]_A \hat{+} \left[\boldsymbol{f}_A^C\right]_A \hat{+} \cdots \hat{+} \left[\boldsymbol{f}_A^Z\right]_A$;
    $\boldsymbol{Party}_Z$ sends $[\sum \boldsymbol{f}_A]_A$ to $\boldsymbol{Party}_A$;
    $\boldsymbol{Party}_A$ decrypts $[\sum \boldsymbol{f}_A]_A$ and identifies $\boldsymbol{A}_{i\in Sky(\boldsymbol{DS}_A)}$ as the global skyline object, if $\sum f_{A,i} = 0$;
end if
```

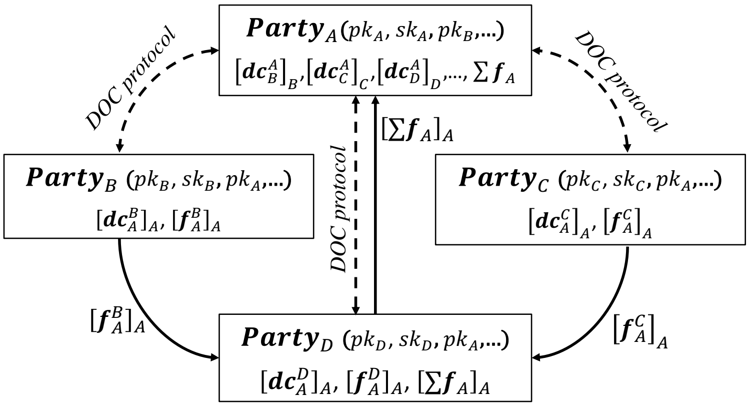

$\boldsymbol{Party}_A (pk_A, sk_A, pk_B, \ldots)$
$[\boldsymbol{dc}^A_B]_B, [\boldsymbol{dc}^A_C]_C, [\boldsymbol{dc}^A_D]_D, \ldots, \Sigma \boldsymbol{f}_A$
DOC protocol
DOC protocol
$[\Sigma \boldsymbol{f}_A]_A$
$\boldsymbol{Party}_B (pk_B, sk_B, pk_A, \ldots)$
$[\boldsymbol{dc}^B_A]_A, [\boldsymbol{f}^B_A]_A$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

Figure 3.3: Data-flow diagram of the MPS protocol for more than two parties

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.*, $[\sum \boldsymbol{f}_A]_A$) according to Step 8 of Algorithm 3, and sends it to $\boldsymbol{Party}_A$. After receiving, $\boldsymbol{Party}_A$ decrypts $[\sum \boldsymbol{f}_A]_A$, and identifies each $\boldsymbol{A}_{i \in Sky(\boldsymbol{DS}_A)}$ as the global skyline object **if** $\sum f_{A,i} = 0$.

Table 3.4: Example of the MPS protocol considering four parties

	$\boldsymbol{Party}_B$			$\boldsymbol{Party}_C$			$\boldsymbol{Party}_D$				$\boldsymbol{Party}_A$
id	$[\boldsymbol{dc}^B_A]_A$	$\boldsymbol{r}$	$[\boldsymbol{f}^B_A]_A$	$[\boldsymbol{dc}^C_A]_A$	$\boldsymbol{r}$	$[\boldsymbol{f}^C_A]_A$	$[\boldsymbol{dc}^D_A]_A$	$\boldsymbol{r}$	$[\boldsymbol{f}^D_A]_A$	$[\sum \boldsymbol{f}_A]_A$	$\sum \boldsymbol{f}_A$
$\boldsymbol{A}_1$	$[1]_A$	12	$[12]_A$	$[2]_A$	6	$[12]_A$	$[0]_A$	8	$[0]_A$	$[24]_A$	24
$\boldsymbol{A}_2$	$[0]_A$	4	$[0]_A$	$[0]_A$	10	$[0]_A$	$[1]_A$	18	$[18]_A$	$[18]_A$	18
$\boldsymbol{A}_3$	$[0]_A$	15	$[0]_A$	$[0]_A$	18	$[0]_A$	$[0]_A$	9	$[0]_A$	$[0]_A$	**0**

Consider four parties ($\boldsymbol{Party}_A$, $\boldsymbol{Party}_B$, $\boldsymbol{Party}_C$, and $\boldsymbol{Party}_D$) want to identify their global skyline objects. Also assume $\boldsymbol{Party}_A$ 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. 3.3 shows a data-flow diagram of the MPS protocol. Besides, Table 3.4 describes the computation results for the local skyline objects of $\boldsymbol{Party}_A$.

3.4 Privacy and Security Analyses

In this section, the privacy and security aspects of the proposed framework is analyzed. According to the composition theorem [11], a framework is considered as secure as long as its elemental protocols are secure, alongside all the intermediate results are random or pseudorandom. Therefore, the privacy and security of the underlying protocols are explained in the following subsections to analyze the privacy and security of this framework.

• **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.

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 $\left[\boldsymbol{dc}_B^A\right]_A$. As a result, by decrypting $\left[\boldsymbol{e}_B^A\right]_A$, $\boldsymbol{Party}_A$ cannot know anything about the local skyline objects of $\boldsymbol{Party}_B$.

• **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 $\boldsymbol{\pi}$ and $\boldsymbol{\pi'}$.

Since $\boldsymbol{Party_A}$ does not know which specific object of $\boldsymbol{Party_A}$ is being compared via the SDC protocol; without knowing $\boldsymbol{M}$, $\boldsymbol{R}$, $\boldsymbol{M'}$, $\boldsymbol{R'}$, $\boldsymbol{\sigma}$, $\boldsymbol{\sigma'}$, $\boldsymbol{\pi}$, and $\boldsymbol{\pi'}$, $\boldsymbol{Party_A}$ 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.

• **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 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.

• **Security of the Proposed Framework:** The proposed framework also maintains the security of the datasets of all participating parties. Within Fig. 3.1, Fig. 3.2, and Fig. 3.3, it can be observed that all the exchanged data are being encrypted before transmission between the parties. Therefore, even if an external adversary or an intruder eavesdrops on the communication media to obtain the transmitted data, it cannot get anything from the encrypted content.

3.5 Performance Evaluation

This section analyzes the complexity and evaluates the performance of the proposed framework. Also, a comparison of the proposed framework with the most relevant work is also presented in Subsection 3.5.3.

3.5.1 Complexity Analyses

Table 3.5 summarizes the required notations for the complexity analyses. Now the analyses of computation and communication complexity of DOC, SDC, and MPS protocols are presented in Table 3.6, Table 3.7, and Table 3.8, respectively.

Table 3.5: Notations for complexity analyses of the multi-party skyline query

Notation	Definition
$\boldsymbol{T_E}$	Complexity of homomorphic encryption
$\boldsymbol{T_D}$	Complexity of homomorphic decryption
$\boldsymbol{T_+}$	Complexity of homomorphic addition
$\boldsymbol{T_\times}$	Complexity of homomorphic multiplication
$\boldsymbol{T_-}$	Complexity of homomorphic subtraction
$\boldsymbol{T_\pi}$	Complexity of vector permutation function
$\boldsymbol{T_H}$	Complexity of hashing function
$\boldsymbol{M}$	Number of parties for multi-party skyline query
$\boldsymbol{N}_A$	Number of $\boldsymbol{Party}_A$'s local skyline objects
$\boldsymbol{N}_B$	Number of $\boldsymbol{Party}_B$'s local skyline objects
$\boldsymbol{B_H}$	Size of encrypted data
$\boldsymbol{B_\#}$	Size of hash data

3.5.2 Experiment

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

Table 3.6: Complexity of the DOC protocol (based on Algorithm 1)

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

* $\boldsymbol{T_{SDC}}$: Total complexity of SDC protocol (Table 3.7)

Table 3.7: Complexity of the SDC protocol (based on Algorithm 2)

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

author builds the program using Java Remote Method Invocation (RMI) framework and uses an 80-bit Paillier encryption key. The author generates synthetic datasets for the experiment where each attribute value of the synthetic datasets is randomly picked from 32-bit unsigned integer.

Initially, the author extracts two sets of local skyline objects from the generated datasets to represent the local skyline objects of two parties. After that, the author examines the effect of dominance comparison through the SDC protocol within the DOC protocol. Since the number of dominance comparisons within the DOC protocol depends on the number of two parties' local skyline objects, the author varies the number of both parties' local skyline objects during the experiment. The author also varies the object dimension from 2

Table 3.8: 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 $\boldsymbol{Party_A}$	
Step 2	$(\boldsymbol{M} - 1) \cdot \boldsymbol{n}_A \cdot \boldsymbol{T}_\times$
	For two parties (Step 4-5)
Step 4	$\boldsymbol{N}_A \cdot \boldsymbol{B_H}$
Step 5	$\boldsymbol{N}_A \cdot \boldsymbol{T_D}$
	For more than two parties (Step 7-10)
Step 7	$(\boldsymbol{M} - 2) \cdot \boldsymbol{N}_A \cdot \boldsymbol{B_H}$
Step 8	$(\boldsymbol{M} - 2) \cdot \boldsymbol{N}_A \cdot \boldsymbol{T}_+$
Step 9	$\boldsymbol{N}_A \cdot \boldsymbol{B_H}$
Step 10	$\boldsymbol{N}_A \cdot \boldsymbol{T_D}$

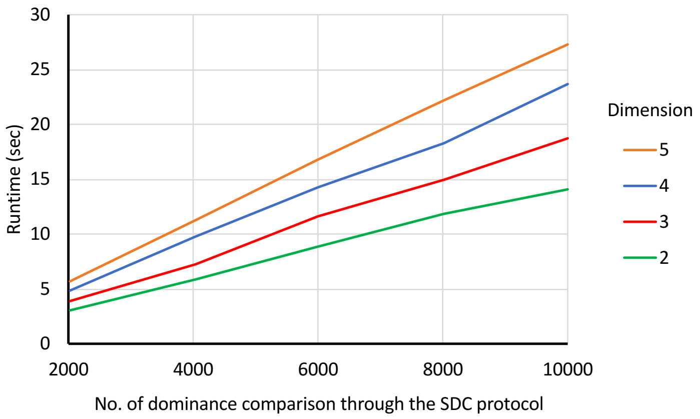


Figure 3.4: Runtime of the DOC protocol

to 5. Based on these, Fig. 3.4 shows the runtime of the DOC protocol. From the figure, it is seen that the runtime is linearly proportional to the number of dominance comparisons through the SDC protocol as well as the number of object dimensions, 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, the author does not evaluate the runtime of the MPS protocol.

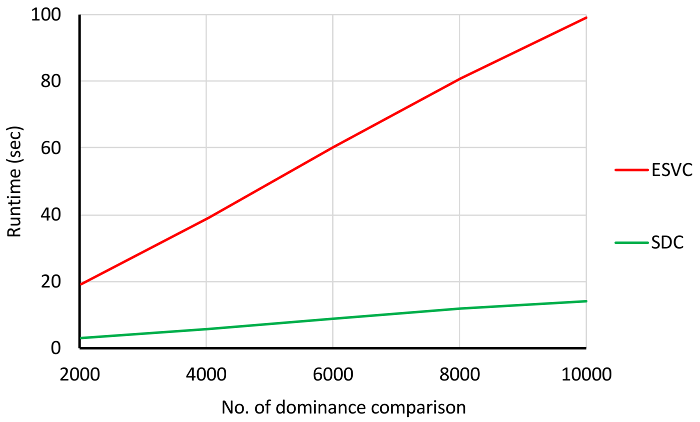


Figure 3.5: Runtime comparison of the proposed SDC protocol with the ESVC protocol [29]. Object dimension: 2, Attribute value length: 32-bit

3.5.3 Comparison

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 dataset in plaintext space. Therefore, it significantly reduces the complexity of multi-party computation.

From the study, it is found that only one framework [29] can compute the privacy-preserving multi-party skyline without incorporating any semi-honest trusted third party. For this reason, the author compares with this one. The ESVC protocol proposed in [29] 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, the data anonymization schemes within the SDC protocol substitute the secure integer comparison. Thereby, the complexity of the SDC protocol does not depend on the attribute value length. Also, the ESVC protocol requires five rounds of data exchanges, whereas the SDC protocol requires only two rounds of data exchanges during secure dominance comparison. Thus, the SDC protocol is more efficient than that of the ESVC protocol. To compare the performance, the author simulate both protocols for the dominance comparison of the two-dimensional dataset objects. Fig. 3.5 shows the runtime comparison of the proposed SDC protocol with the ESVC protocol. From the figure, it can be seen that the runtime of the ESVC protocol is much higher than the SDC protocol.

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

3.6 Concluding Remarks

This chapter presents 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 algorithms and data-flow diagrams of the underlying protocols, confirms that all participating organizations can recognize their multi-party skyline objects without revealing their dataset to others. Since this work does not incorporate any third party, this model do not rely on the credibility of the third party. The privacy and security analyses demonstrate that the framework satisfies the desired privacy requirements. Also, the proposed framework achieves significant efficiency and security by the avoidance of 'secure integer comparison', and the exploitation of encryption of 'the dominance comparison result' within the SDC protocol. The efficiency of the proposed framework for real-world deployment is shown through the extensive performance evaluation.

Chapter 4

Privacy-preserving Multi-party K-Skyband Query

This chapter proposes an efficient framework for privacy-preserving multi-party K-skyband query. The proposed framework can compute the multi-party K-skyband without disclosing the objects in a party to other parties, which is essential in privacy-aware applications. This framework introduces a novel method for transforming objects' attributes without changing their sorting order rank on each dimension of the object in a privacy-preserving multi-party computation environment to improve the computation efficiency. After that, the sorting order rank of the objects' attributes on each dimension is utilized for the multi-party K-skyband computation. The proposed multi-party objects' attributes' transformation method exploits the Paillier cryptosystem and its properties in the semi-honest adversary model. The author also analyzes the privacy and security of the proposed scheme and evaluates its performance in the real environment. The experimental results show that the proposed scheme is highly efficient in terms of computation complexity.

The rest of this chapter is organized as follows: Section 4.1 and 4.2 briefly explain the proposed system model and the desired privacy requirements, respectively. Then, Section 4.3 specifies the detail of the proposed framework. The proposed framework is discussed in Section 4.4. After that, the efficiency of the proposed framework is presented concerning computational and communication complexity, and process execution time in Section 4.5, followed by the conclusion in Section 4.6. The hexadecimal number system is used throughout this chapter for a better understanding of the proposed framework.

4.1 System Model

Since the participating parties never disclose their objects to others, it is considered to securely transform the value of objects' attributes without changing their sorting order on each dimension. The system includes two semi-honest third parties to construct the objects' attributes sorting order in a secure way and to compute the multi-party K-skyband from the transformed sorting order values of the objects. The semi-honest third parties are the coordinator and the Substitution Vector Constructor(SVC). These are considered to be honest-but-curious parties. It is assumed that all participating parties, along with the coordinator and the SVC, execute the protocol strictly. However, they intend to extract the private data of others from the computation. Therefore, the author considers that the objects' attributes order construction process should need to be secured enough so that nothing could be obtained by the coordinator other than the transformed sorting order value of objects' secret attributes on each dimension of the object.

Coordinator: The coordinator initiates the multi-party K-skyband computation process in collaboration with all participating parties. At first, the coordinator constructs the homomorphic encryption key pair for protecting the transmitted data through the communication linkage. After that, the coordinator distributes the public encryption key among the participating parties along with the SVC. The coordinator also computes the multi-party K-skyband objects from the transformed sorting order value of the multi-party objects' attributes.

Substitution Vector Constructor (SVC): The SVC generates the substitution vectors for the participating parties to transform the objects' attributes without changing the objects' sorting order on each dimension. To secure the substitution vectors, the SVC encrypts each element of the substitution vectors using public encryption key provided by the coordinator before distributing it to the participating parties. Due to encryption of the substitution vectors, if any external intruder eavesdrops on the communication linkage, it cannot get anything from the encrypted data, while the participating parties can still be able to transform the objects' attributes by using the encrypted substitution vectors.

4.2 Desired Privacy

In the proposed system model, it is mainly considered to extract multi-party K-skyband objects from the secure multi-party databases in an efficient and privacy-preserving way. Since an external intruder may eavesdrop on the communication linkage of the participating parties, the author considers encrypting the substitution vectors and the transformed sorting order values of the objects' attributes using homomorphic encryption while transferring the data between the entities. Therefore, even if an external intruder eavesdrops on the communication linkage and obtains the transmitted data, it cannot get anything from the encrypted data.

On the other hand, if the coordinator gets the encrypted substitution vectors from any participating party or the SVC, the coordinator may restore the original objects from the transformed sorting order of the objects' attributes on each dimension. So, the proposed system model strictly assumes that any participating party and the SVC do not provide the encrypted substitution vectors to the coordinator.

4.3 Proposed Framework

As described in Section 4.1, there are three groups of entities for the proposed secure multi-party K-skyband computation framework - first, the participating organizations/parties with secured databases. Then secondly, the coordinator, to whom the participating parties send the transformed sorting order values of the objects' attributes on each dimension. And finally, the SVC, who construct the encrypted substitution vectors for the participating parties to transform the objects' attributes. Figure 4.1 illustrates the proposed framework.

According to Figure 4.1, the coordinator constructs the homomorphic encryption key pair using the Paillier cryptosystem. After that, the coordinator distributes the public encryption key to the SVC and each participating party. After getting the public encryption key from the coordinator, the SVC generates the substitution vectors and distributes the encrypted substitution vectors among the participating parties. At the same time, each party computes K-skyband objects from its local database. Later, each party uses the encrypted substitution vectors supplied by the SVC to construct the encrypted sorting order of the secured objects' attributes on each dimension of the object. After that, each party sends the encrypted sorting order values of local K-skyband objects' attributes to the coordinator. The coordinator decrypts the transformed sorting order values and computes the multi-party K-skyband from the transformed sorting order values of each party's local K-skyband objects' attributes.

4.3.1 Substitution Vector Construction

It is admitted that the numerical value of the objects' attribute could be significantly longer. However, creating a large substitution vector, and distributing the large vector among multiple parties for substituting the objects' attributes is not computationally effective. For example, to convert a 32-bit attribute value, a substitution vector of size 2^{32} is required, which is significantly large. Therefore, the author considers splitting the attribute value

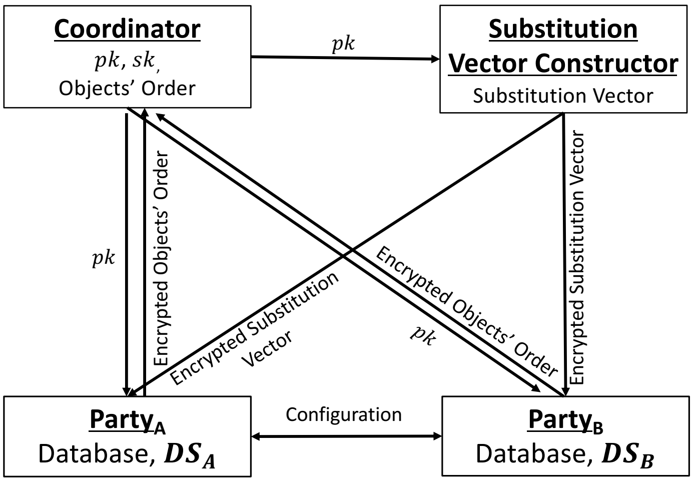


Figure 4.1: System diagram of the multi-party K-skyband query framework

into multiple bit-slices and creating substitution vectors according to the length of bit-slices. During the transformation of the objects' attributes, the index of substitution vector elements can be determined by the numerical values of the split slices of the object's attribute. In the proposed framework, the SVC creates encrypted substitution vectors for transforming the objects' attribute value into their encrypted sorting order.

The SVC creates substitution vectors for variant bit-slice length from 4 to 10 so that the participating parties can decide bit-slice length for splitting the attribute value without communicating with SVC. It is also admitted that SVC creates multiple substitution vectors for each bit-slice length. Creating substitution vectors considering variant bit-slice length and multiple vectors for each bit-slice length, is essential for constructing anonymize sorting order values of the objects' attributes on each dimension of the object.

To create a substitution vector for bit-slice length n, at first, SVC generates 2^n random integers between 2^{n-1} to $(2^n - 1)$. After that, SVC computes the cumulative summation of the generated random numbers. Finally, SVC encrypts the cumulative sums using the Paillier public key supplied by the coordinator to create an encrypted substitution vector for bit-slice length n. An example of substitution vector construction for bit-slice length $n = 4$ is illustrated in Table 4.1. Within Table 4.1, $V_{4,1,7}$ denotes that it is the 7^{th} encrypted

Table 4.1: Example of Substitution Vector for $\boldsymbol{n} = 4$

	Random Number	**Cumulative Sum**	**Encrypted Sum**
i	r_i	ξ_i	$V_i = [\xi_i]$
0	E	E	$V_{4,1,0}$
1	C	1A	$V_{4,1,1}$
2	9	23	$V_{4,1,2}$
3	C	2F	$V_{4,1,3}$
4	F	3E	$V_{4,1,4}$
5	D	4B	$V_{4,1,5}$
6	A	55	$V_{4,1,6}$
7	A	5F	$V_{4,1,7}$
8	E	6D	$V_{4,1,8}$
9	B	78	$V_{4,1,9}$
A	8	80	$V_{4,1,A}$
B	9	89	$V_{4,1,B}$
C	D	96	$V_{4,1,C}$
D	F	A5	$V_{4,1,D}$
E	8	AD	$V_{4,1,E}$
F	B	B8	$V_{4,1,F}$

vector element of the 1^{st} substitution vector table generated for the bit-slice length 4.

4.3.2 Computation by Individual Parties

It is considered that, at first, each participating party computes the K-skyband objects from its local database for enhancing the efficiency of secure multi-party K-skyband computation from the transformed sorting order value of the objects' attributes. The local K-skyband computation by each participating party can also reduce the risk of secure database disclosure by analyzing the sorting order values of the objects' attributes by the coordinator.

After computing the local K-skyband, each party transforms the attributes of the local K-skyband objects into its encrypted sorting order on each dimension using the encrypted substitution vectors supplied by SVC. When all participating parties send the encrypted sorting order values of the local K-skyband objects' attributes to the coordinator, the coordinator can decrypt sorting order values of the local K-skyband objects' attributes using its private decryption key. Then, by using the decrypted sorting order values of the objects' attributes, the coordinator can compute the multi-party K-skyband. Due to the additivity property of K-skyband computation, it can be said that computed multi-party K-skyband objects remain equal as the K-skyband objects computed from the union of each party's objects.

To transform the object's attributes, all participating parties mutually fix the bit-slice length for splitting the attribute value on each dimension of the local K-skyband objects. The bit-slice length for splitting the attribute value is denoted by $n_i\,(i = 0, 1, 2, \cdots)$, where n_0 represents the length of less significant bit-slice of the attribute value. Since the SVC provides multiple substitution vectors for each bit-slice length, all participating parties also mutually determine the specific encrypted substitution vector for every bit-slice of the attribute value. It is required for ensuring that the transformed sorting order value of the objects' attributes remain identical for all parties. The coordinator may try to detect any pattern from the transformed sorting order values of the objects' attributes. Therefore, any participating party will not provide the bit-slice length to the coordinator to prevent the coordinator from restoring the objects from the objects' sorting order values on each dimension of the objects.

After substituting the bit-slices of each attribute value by encrypted substitution vector elements, each party uses the encrypted substitution vector elements of each secured attribute value to build the encrypted sorting order value. Each party utilizes the homomorphic addition and multiplication properties of the Paillier cryptosystem to construct encrypted sorting order values of the objects' attributes. A constant integer will be multiplied with the substitution vector elements (except the substitution vector element of the less significant bit-slice) so that the transformed values of the objects' attributes follow the same sorting order as the original attribute value. The constant multiplier 2^{m_i} needs to be chosen in such a way, for which the transformed values do not change the sorting order of the original objects' attributes. It should also need to be considered that the coordinator cannot get any pattern by splitting the transformed sorting order values of the objects' attributes.

Since SVC generates the random numbers between 2^{n_i-1} to $2^{n_i} - 1$ for creating the substitution vector of bit-slice length n_i, the minimum difference between any two substitution vector elements is greater than or equal to 2^{n_i-1}. Therefore, for the given bit-slice lengths to split the attribute value, m_i can be computed using the following equation.

$$m_i = n_0 - n_i + \sum_{j=0}^{i-1}(n_j + 1), i = 1, 2, \cdots \tag{4.1}$$

Since it is not possible to multiply a rational number with the encrypted value in Paillier cryptosystem, the participating parties should choose all bit-slice lengths (n_i) in such a way for which the constant multiplier 2^{m_i} will be an integer, i.e., $m_i \geq 0$.

Assume, for transforming an attribute value α into encrypted sorting order value γ, the attribute value α has been split into S slices using bit-slice length $n_0, n_1, \cdots n_{S-1}$. Consider $\alpha_0, \alpha_1, \cdots, \alpha_{S-1}$ represent the bit slices of the attribute value α. If $\beta_0, \beta_1, \cdots, \beta_{S-1}$ represent the encrypted substitution vector elements of $\alpha_0, \alpha_1, \cdots, \alpha_{S-1}$, respectively, the transformed value γ can be determined by equation 4.2.

$$\gamma = \beta_0 + \sum_{i=1}^{S-1}(\beta_i \times 2^{m_i}) \tag{4.2}$$

Note that, in equation 4.2, γ will be computed using homomorphic addition and multiplication properties of the Paillier cryptosystem.

Consider two participating parties: $Party_A$ and $Party_B$, respectively. $Party_A$ has $2SB\,(DS_A)$ as its 2-skyband objects set, while $Party_B$ has $2SB\,(DS_B)$ as its 2-skyband objects set. Table 4.2 describes $2SB\,(DS_A)$ and $2SB\,(DS_B)$.

Table 4.2: 2-skyband objects set of $Party_A$ and $Party_B$

$2SB\,(DS_A)$

ID	$\mathbf{d_1}$	$\mathbf{d_2}$
a	FAB	442
b	262	B6D
c	481	479
d	442	E95
e	ADA	249
f	71E	68F
g	845	90F
h	63F	DF5

$2SB\,(DS_B)$

ID	$\mathbf{d_1}$	$\mathbf{d_2}$
m	CFD	5B0
n	942	532
o	600	823
p	4F3	759
q	543	AB4
r	3C4	C40
s	C7F	380

Let the parties have mutually fixed to split the attribute value of dimension $\mathbf{d_1}$ into three 4-bit slices to transform the attribute value, *i.e.*, $n_0 = n_1 = n_2 = 4$, where n_0 denotes the length of less significant bit-slice and n_2 denotes the length of most significant bit-slice. Also, consider the parties have decided to use the encrypted substitution vector $V_{4,1}$ for substituting less significant bit-slice of the attribute values. Similarly, they have also chosen to use $V_{4,2}$ and $V_{4,3}$ for the other bit-slices of the attribute values.

For example, consider an attribute value belongs to $\mathbf{d_1}$ dimension of object p is $(4\text{F}3)_{\text{hex}}$. The encrypted substitution vector elements of the split slices of $(4\text{F}3)_{\text{hex}}$ are as follow:

$$V_{4,1,3} = [2F]$$

$$V_{4,2,F} = [A7]$$

$$V_{4,3,4} = [37]$$

Therefore, the transformed encrypted sorting order value of object p's $\mathbf{d_1}$ dimension attribute can be computed using equation 4.2.

$$\gamma_{p,1} = V_{4,1,3}\hat{+}(V_{4,2,F}\hat{\times}2^5)\hat{+}(V_{4,3,4}\hat{\times}2^{10}) = [F10F]$$

Table 4.3 and 4.4 describe the encrypted object sorting order construction for each object's attribute on dimension $\mathbf{d_1}$.

Table 4.3: Encrypted sorting order construction by $Party_A$

ID	$\mathbf{d_1}$	**Bit-Slice**			**Substitute Value**			**Encrypted**
		α_2	α_1	α_0	β_2	β_1	β_0	**Order,** γ_1
a	FAB	F	A	B	$V_{4,3,F}$	$V_{4,2,A}$	$V_{4,1,B}$	$\gamma_{a,1}$
b	262	2	6	2	$V_{4,3,2}$	$V_{4,2,6}$	$V_{4,1,2}$	$\gamma_{b,1}$
c	481	4	8	1	$V_{4,3,4}$	$V_{4,2,8}$	$V_{4,1,1}$	$\gamma_{c,1}$
d	442	4	4	2	$V_{4,3,4}$	$V_{4,2,4}$	$V_{4,1,2}$	$\gamma_{d,1}$
e	ADA	A	D	A	$V_{4,3,A}$	$V_{4,2,D}$	$V_{4,1,A}$	$\gamma_{e,1}$
f	71E	7	1	E	$V_{4,3,7}$	$V_{4,2,1}$	$V_{4,1,E}$	$\gamma_{f,1}$
g	845	8	4	5	$V_{4,3,8}$	$V_{4,2,4}$	$V_{4,1,5}$	$\gamma_{g,1}$
h	63F	6	3	F	$V_{4,3,6}$	$V_{4,2,3}$	$V_{4,1,F}$	$\gamma_{h,1}$

Table 4.4: Encrypted sorting order construction by $Party_B$

ID	$\mathbf{d_1}$	**Bit-Slice**			**Substitute Value**			**Encrypted**
		α_2	α_1	α_0	β_2	β_1	β_0	**Order,** γ_1
m	CFD	C	F	D	$V_{4,3,C}$	$V_{4,2,F}$	$V_{4,1,D}$	$\gamma_{m,1}$
n	942	9	4	2	$V_{4,3,2}$	$V_{4,2,4}$	$V_{4,1,2}$	$\gamma_{n,1}$
o	600	6	0	0	$V_{4,3,6}$	$V_{4,2,0}$	$V_{4,1,0}$	$\gamma_{o,1}$
p	4F3	4	F	3	$V_{4,3,4}$	$V_{4,2,F}$	$V_{4,1,3}$	$\gamma_{p,1}$
q	543	5	4	3	$V_{4,3,5}$	$V_{4,2,4}$	$V_{4,1,3}$	$\gamma_{q,1}$
r	3C4	3	C	4	$V_{4,3,3}$	$V_{4,2,C}$	$V_{4,1,4}$	$\gamma_{r,1}$
s	C7F	C	7	F	$V_{4,3,C}$	$V_{4,2,7}$	$V_{4,1,F}$	$\gamma_{s,1}$

In the same way, the participating parties can also construct the encrypted sorting order values for the other dimensions of the objects. After building the encrypted sorting order values for all dimensions of the object, each party sends the encrypted sorting order values of the objects' attributes to the coordinator for computing multi-party K-skyband.

4.3.3 Computation by the Coordinator

After receiving the encrypted sorting order values from all participating parties, the coordinator decrypts the encrypted sorting order values of the objects' attributes by using the private decryption key sk. For the running example, Table 4.5 illustrates the sorting order values of the objects' attributes obtained by the coordinator after decryption.

Table 4.5: Decrypted transformed values of the objects' attributes

ID	**Encrypted sorting order**		**Decrypted Order** $\theta_i = \mathrm{De}_{sk}(\gamma_i)$	
	γ_1	γ_2	θ_1	θ_2
a	$\gamma_{a,1}$	$\gamma_{a,2}$	2FEE9	EB62
b	$\gamma_{b,1}$	$\gamma_{b,2}$	9D23	24F4D
c	$\gamma_{c,1}$	$\gamma_{c,2}$	E75A	F075
d	$\gamma_{d,1}$	$\gamma_{d,2}$	E2E3	2E368
e	$\gamma_{e,1}$	$\gamma_{e,2}$	21A80	8BB5
f	$\gamma_{f,1}$	$\gamma_{f,2}$	16ACD	155FF
g	$\gamma_{g,1}$	$\gamma_{g,2}$	19F0B	1DDDF
h	$\gamma_{h,1}$	$\gamma_{h,2}$	141D8	2AF48
m	$\gamma_{m,1}$	$\gamma_{m,2}$	26D85	11DEB
n	$\gamma_{n,1}$	$\gamma_{n,2}$	1D2E3	11262
o	$\gamma_{o,1}$	$\gamma_{o,2}$	13D2E	1ACF1
p	$\gamma_{p,1}$	$\gamma_{p,2}$	F10F	18575
q	$\gamma_{q,1}$	$\gamma_{q,2}$	106EF	21E1E
r	$\gamma_{r,1}$	$\gamma_{r,2}$	CCDE	26F4B
s	$\gamma_{s,1}$	$\gamma_{s,2}$	262F8	CD4B

Then the coordinator obtains the number of dominants of each party's local K-skyband object by comparing the transformed sorting order values of the objects' attributes. When the number of dominants of an object is less than or equal to K, the coordinator identifies the object as a multi-party K-skyband object. Table 4.6 describes the number of dominants of each party's local K-skyband object computed by comparing the transformed sorting order values of the objects' attributes.

Later, the coordinator sends the *ID*s of the multi-party K-skyband objects to the corresponding participating party.

4.4 Privacy and Security Analyses

The proposed framework of secure multi-party K-skyband computation is based on transforming the objects' attributes without changing the order of the objects' attributes on

Table 4.6: Multi-party K-skyband query from the objects' attributes' transformed value

ID	**Decrypted Order**		**No. of**
	θ_1	θ_2	**Dominant Objects**
a	2FEE9	EB62	2
b	9D23	24F4D	0
c	E75A	F075	0
d	E2E3	2E368	2
e	21A80	8BB5	0
f	16ACD	155FF	1
g	19F0B	1DDDF	3
h	141D8	2AF48	3
m	26D85	11DEB	3
n	1D2E3	11262	1
o	13D2E	1ACF1	2
p	F10F	18575	1
q	106EF	21E1E	2
r	CCDE	26F4B	1
s	262F8	CD4B	1

each dimension of the object. The author considers constructing the encrypted substitution vectors to achieve the transformed sorting order values of the secured multi-party objects' attributes. As a semi-honest adversary model, this framework implicitly assumes that the SVC, any participating party, and the coordinator do not collude with each other. The proposed framework strictly assumes that the coordinator does not make any secret alliance with any dishonest party to get the encrypted substitution vector and the length of bit-slices used for transforming the objects' attributes.

In this framework, only the coordinator has the private decryption key. Hence, any participating party or any external intruder cannot decrypt the encrypted substitution vector and the encrypted sorting order values of the objects' attributes. As a result, only the coordinator can achieve the transformed sorting order values of the objects' attributes after decryption.

Since the participating parties only share the encrypted attributes sorting order values of their local K-skyband objects with the coordinator, it cannot be possible for the coordinator to guess the original attributes by analyzing the frequency of the limited number of transformed sorting order values of the objects' attributes. However, if any dishonest party or the SVC provides the encrypted substitute vectors and the bit-slice length to the coordinator, the coordinator can restore the objects from the transformed sorting order values of the objects' attributes. In such a case, the proposed framework cannot fulfill the necessary privacy and security requirements.

Therefore, it can be said that, as long as the coordinator is honest, the proposed framework can assure the objects' privacy and security during the multi-party K-skyband computation.

4.5 Performance Evaluation

According to the explanation, it can be said that the proposed framework of secure-preserving multi-party K-skyband computation will produce the correct result for all positive integer attribute value. This section describes the experimental results to examine the efficiency of the proposed method. The author also comprehensively compared the complexity of the proposed framework with the frameworks proposed in [29] and [44].

4.5.1 Experiment

The author evaluates the efficiency of the proposed technique using four identical computers with Intel® Core i5 6500 3.20 GHz CPU, 8GB DDR3 memory, and 64-bit Ubuntu 16.04 OS. Out of those four computers, one is considered as the coordinator, one is considered as the Substitution Vector Constructor (SVC), and the other two computers is considered as the individual parties with private databases. The author compiles the source codes with the Java V8 compiler to implement the proposed framework for performance evaluation. The author checked the process running time of the proposed framework for computing multi-party K-skyband from secure multi-party databases.

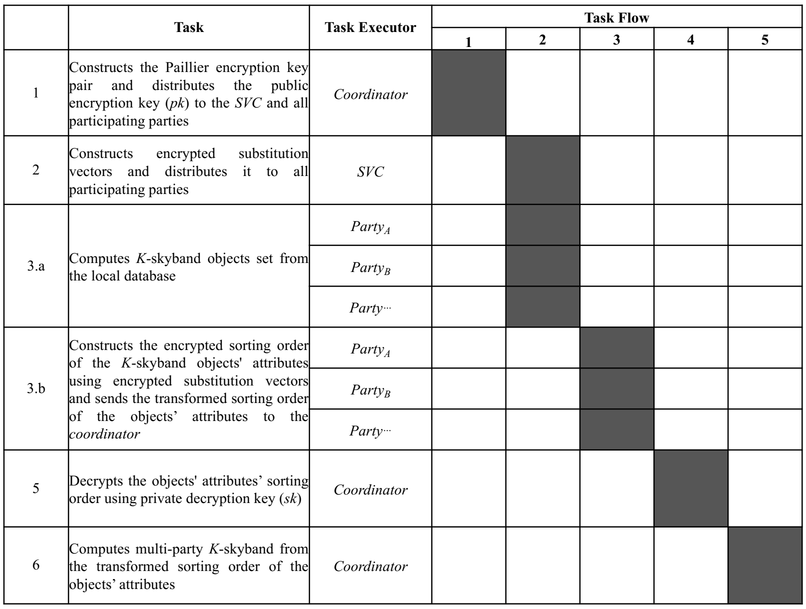
The author created the synthetic datasets for evaluating the performance of the proposed framework, where each objects' attribute of the synthetic datasets was randomly picked from 0 to 2^{31}. The experimental design aim to check the computation overhead of the proposed K-skyband computation approach with the varied value of K, varied object dimension, and the varied number of database objects obtained by the participating parties. For the experiment, it is considered that each participating party has the same amount of database objects from which each party computes the local K-skyband objects.

For evaluating the efficiency of the framework, it is considered that, after the initialization by the coordinator by distributing the Paillier public encryption key, the participating parties begin the process of computing local K-skyband. At the same time, the SVC starts the process of creating and distributing encrypted substitution vectors. After obtaining the encrypted substitution vectors from the SVC, the participating parties transform the attributes of local K-skyband objects into their encrypted sorting order values. The Gantt-chart of Figure 4.2 describes the task execution flow of the proposed framework.

The author uses the 80-bit Paillier encryption key for the experiment. Besides that, the author splits the 32-bit objects' attributes on each dimension of the object into four 8-bit slices for substituting the attributes of the objects using substitution vectors.

• **The effect of K for K-skyband query:** During the experiment for the performance evaluation of the proposed secure multi-party K-skyband computation framework, the author studied the process execution runtime for varied K, at first. Figure 4.3 describes the entire process running time for different value of K. It is known that, the number of comparisons, the quantity of local skyband objects, and the number of homomorphic encryption operations are proportional to K for the K-skyband query. Therefore, the process running time varies with K. The experimental results also show that consequence.

• **Process running time with respect to the dimension of the object:** Figure 4.4 illustrates the effect of object dimension for computing K-skyband. Since the number



	Task	Task Executor	Task Flow				
			1	2	3	4	5
1	Constructs the Paillier encryption key pair and distributes the public encryption key (*pk*) to the *SVC* and all participating parties	*Coordinator*	■				
2	Constructs encrypted substitution vectors and distributes it to all participating parties	*SVC*		■			
3.a	Computes *K*-skyband objects set from the local database	*Party*$_A$		■			
		Party$_B$		■			
		Party$_{\cdots}$		■			
3.b	Constructs the encrypted sorting order of the *K*-skyband objects' attributes using encrypted substitution vectors and sends the transformed sorting order of the objects' attributes to the *coordinator*	*Party*$_A$			■		
		Party$_B$			■		
		Party$_{\cdots}$			■		
5	Decrypts the objects' attributes' sorting order using private decryption key (*sk*)	*Coordinator*				■	
6	Computes multi-party *K*-skyband from the transformed sorting order of the objects' attributes	*Coordinator*					■

Figure 4.2: Task-flow of the multi-party K-skyband query framework

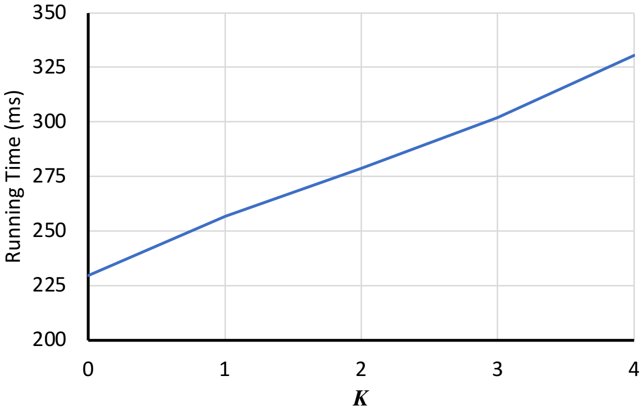


Figure 4.3: Runtime of the multi-party K-skyband query for varied K
[Number of tuples: 25000/Party; Dimension of the object: 4]

of encrypted substitution vector along with the number of comparisons and the number of locally computed K-skyband objects increases with the object dimension, the process execution time also increases. The results of the experiment also reflect it.

• **Process running time with respect to the number of tuples:** The author observed the effect of the number of tuples on the process running time. For this experiment, the author also varied the object dimension along with the number of tuples for this experiment. Figure 4.5 reports the experimental result for evaluating the performance of the proposed

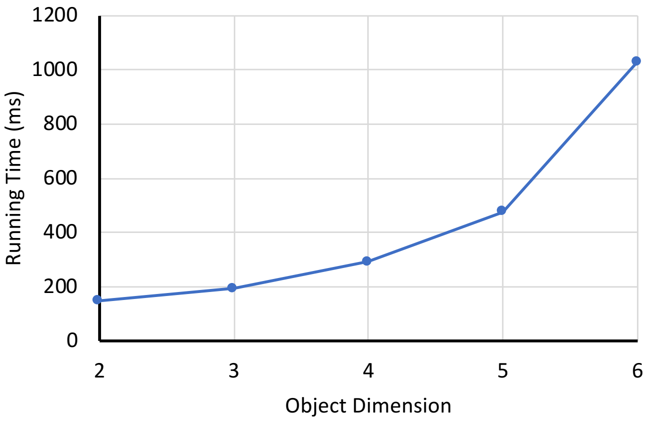


Figure 4.4: Runtime of the multi-party K-skyband query for varied object dimension [Number of tuples: 25000/Party, $K = 2$]

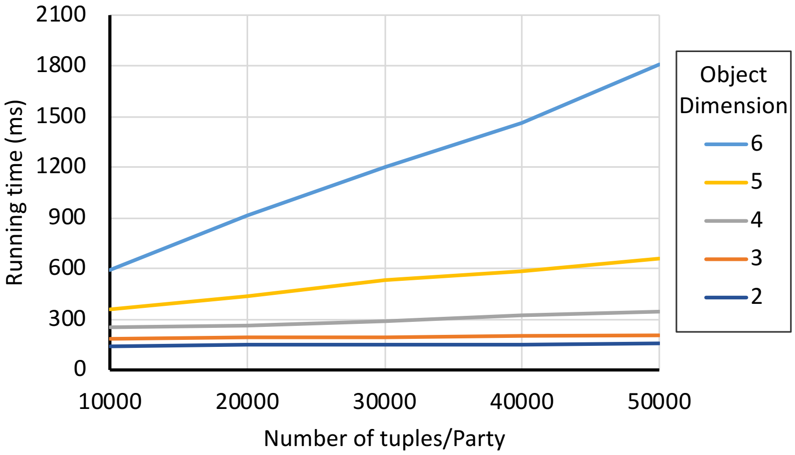


Figure 4.5: Runtime of the multi-party K-skyband query for varied tuples [$K = 2$]

framework for the varied number of tuples per party and different object dimension.

4.5.2 Comparison

The framework proposed in [29] requires five rounds of data transmission between two parties for comparing the dominance relation between an object in one party's dataset with an object in another party's dataset. Furthermore, it is necessary to create the homomorphic encryption key twice for each secure dominance comparison. Besides that, for comparing the dominance relation between two objects, this framework utilizes the 0-encoding and 1-encoding scheme of Lin et al. [23], while the computation and communication complexity of 0-encoding and 1-encoding scheme depends on the maximum range of the integer attribute value.

On the other hand, the framework proposed in [44] does not require such a secure dominance comparison. However, it requires several rounds of data transmission and computation between the participating parties and the coordinator to construct the sorting order of the multi-party objects' attributes on each dimension of the objects securely. Moreover, the performance of this framework also depends on the maximum range of the attribute value.

In comparison with the above frameworks, the complexity of the proposed framework does not depend on the maximum range of the integer attribute value. Since the proposed framework of secure multi-party K-skyband computation is based on secure objects order construction, it does not require computationally expensive secure object dominance comparison like [29]. Furthermore, the proposed framework also does not require multiple rounds of data transmission and computation for constructing objects' attributes' sorting order like [44]. Therefore, it can be said that the proposed algorithm is more efficient and robust in terms of computation and communication complexity.

4.6 Concluding Remarks

The proposed framework addresses the data privacy and security issues of the K-skyband query in distributed multi-party databases. To maintain data privacy, the proposed framework considered to transform the multi-party objects' attributes without altering their sorting order rank on each dimension. Since the K-skyband query requires more object dominance comparison than the skyline query, the secure dominance comparison based multi-party K-skyband query will consume significant time for the secure computation. However, the proposed framework does not require such secure dominance comparison. Therefore, it can be said that the framework is more effective compared to others.

Chapter 5

Privacy-preserving Multi-party Top-k Dominating Query

Recently, preference-based queries have drawn massive attention in the database community. Especially, the top-k query has gained notable importance, which retrieves the k data objects that have better scores than others based on user-defined monotone scoring function. However, it is not easy to specify an appropriate scoring function for selecting top-k multidimensional objects from a database. As a variant of the skyline query, the top-k dominating query can be used for this purpose. It returns the top-k objects based on the 'domination score', which can be calculated without a user-specified scoring function.

In many cases, multiple organizations, which are running similar trades and maintaining comparable databases, want to recognize the top-k dominating objects from the union of their databases. Such recognition of the multi-party top-k dominating query can help the organizations to locate their most competitive products or services. Since the database contains sensitive information about the products and services, any organization does not want to reveal its private database to others. On the other hand, it is not possible to compute the 'domination score' of the database objects without revealing the objects to others.

In this chapter, the author addresses this problem and proposes a framework for computing the multi-party top-k dominating query in the combined multi-party databases. In this proposed framework, organizations/parties will not be required to reveal their secured database objects to others. Besides, the framework also ensures that only the participating parties can recognize their qualified multi-party top-k dominating objects. Although some algorithms for privacy-preserving skyline and top-k queries in the distributed multi-party databases were proposed, to the best of our knowledge, there are no studies that deal with the issue of data privacy and security for top-k dominating queries in distributed multi-party databases.

Here the author first describes the system model and the desired privacy requirements in Section 5.1 and Section 5.2. Section 5.3 introduces some of the basic security sub-protocols that are utilized in the proposed framework, followed by the specification of detailed algorithms of the proposed framework in Section 5.4. Then the privacy and security analyses and the performance evaluation of the proposed framework are discussed in Section 5.5 and Section 5.6, respectively. Finally, Section 5.7 concludes this chapter.

5.1 System Model

The proposed framework adapts the semi-honest adversary computation model. It consists of two groups of entities: the participating parties or agents with private datasets and two trusted semi-honest non-colluding third parties, namely, the Encryption Key Service Provider (EKSP) and the Encrypted Data Repository Service (EDRS) Provider. Fig. 5.1 illustrates the proposed system model.

Here the participating parties or the agents are the entities; those have private datasets and want to recognize their multi-party top-k dominating objects. The EKSP generates the key-pair of Paillier cryptosystem and distributes the public key to the participating parties as well as the EDRS. The EKSP also actively participates during the secure computation of

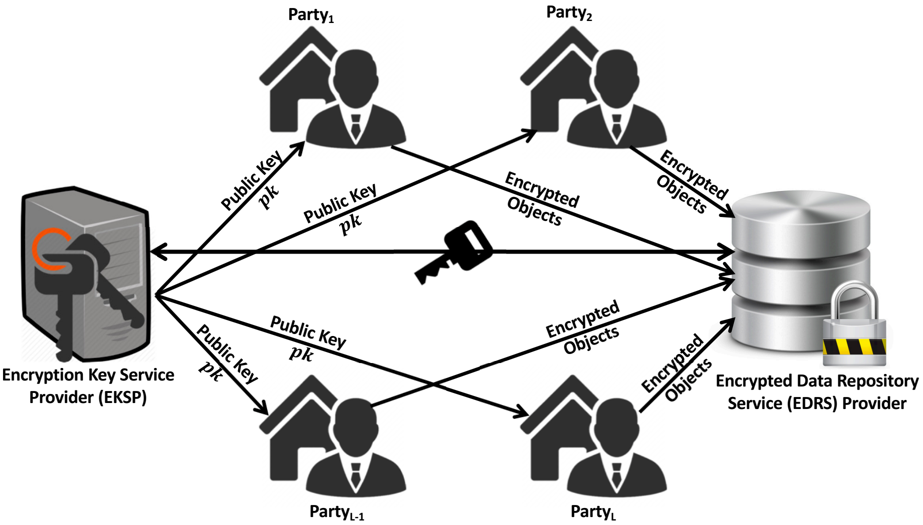


Figure 5.1: System model of the multi-party top-k dominating query

the multi-party top-k dominating query. The EDRS stores the encrypted multi-dimensional dataset supplied by the participating parties. It also preserves the encrypted intermediate results during the privacy-preserving multi-party top-k dominating query.

5.2 Desired Privacy

In the proposed system model the participating parties, the EKSP, and the EDRS are assumed to be honest-but-curious entities in the sense that they follow the proposed protocols strictly. An external adversary may eavesdrop on the communication media to obtain the transmitted content during the secure computation process. Therefore, the proposed system model must ensure that any external adversary cannot know anything from the transmitted content. It also assume that an adversary may compromise either the EKSP or the EDRS. However, such an adversary is restricted from compromising both the EKSP and the EDRS concurrently. It is observed that these kinds of restrictions are common in the adversary models utilized in cryptographic protocols. Besides, it is admitted that the EKSP and the EDRS could not know anything about the query result, *e.g.*, which objects are in the multi-party top-k dominating query result.

5.3 Basic Security Sub-protocols

This section presents three secure sub-protocols that will be used within the proposed framework. All protocols assume a two-party setting: the EDRS has the encrypted data, and the EKSP has the private key sk. Through these sub-protocols, the EDRS obtains an encrypted result of a function on the encrypted data without disclosing the original encrypted data to either the EDRS or the EKSP. For each sub-protocol, this section briefly describes their input(s) and output, and their detail structures are available in [8].

• **Secure Multiplication (SM) Protocol:** Assume the EDRS has the encrypted input $[a]$ and $[b]$, and the EKSP has the private key sk, where a and b are not known to both of them. The Secure Multiplication (SM) protocol securely computes the encrypted result $[c] := [a \times b]$, such that, only the EDRS obtains $[c]$ and no information related to a, b, and c are revealed to the EDRS or the EKSP.

• **Secure Minimum (SMIN) Protocol:** Assume the EDRS has the encrypted input $[a]$ and $[b]$, and the EKSP has the private key sk, where a and b are not known to both of them. Then the $SMIN$ protocol securely computes the encrypted minimum value of a and b, *e.g.*, $[c] := SMIN\left([a], [b]\right)$, such that, only the EDRS receives $[c]$.

• **Secure Minimum out of n Numbers ($\mathbf{SMIN_n}$) Protocol:** Assume the EDRS has the array of n encrypted input $\left([d_1], \cdots, [d_n]\right)$ and the EKSP has the private key sk. The goal of $SMIN_n$ protocol is to securely compute the encrypted minimum value of this n inputs. The $SMIN_n$ protocol utilizes $SMIN$ protocol for computing the encrypted minimum of two encrypted inputs securely. Through this protocol, the EDRS obtains the encrypted minimum value and no information is revealed to the EDRS or the EKSP.

5.4 Proposed Framework

In this section, the proposed framework for the privacy-preserving multi-party top-k dominating query is described. The following computation phases are considered for the proposed framework.

1. The EKSP generates the homomorphic encryption key pair (pk, sk) and distributes the public encryption key pk to others.

2. Every participating party prepares the dataset for multi-party top-k dominating query.

3. The EDRS and the EKSP jointly compute the top-k dominating query from the encrypted multi-party datasets.

4. Every participating party recognizes its multi-party top-k dominating objects.

• **The encryption key pair generation:** At the very beginning of the computation, the EKSP generates the key pair of Paillier cryptosystem. The detailed key generation process is explained in [31]. After generating the key pair, the EKSP distributes the public encryption key pk to all participating parties and the EDRS.

• **Computation by the individual participating party:** To enhance the computation efficiency, every party firstly computes the μ-score of every object of its dataset in the proposed framework. According to the property of the top-k dominating query, it is known that the top-k dominating objects belong to $(k-1)$-skyband objects. Besides, every multi-party $(k-1)$-skyband object is also a $(k-1)$-skyband object of any participating party's dataset. Therefore, every party also computes $(k-1)$-skyband query during the computation of the μ-score of its dataset objects. Such prior computation reduces the computation complexity of the multi-party top-k dominating query by reducing the number of dominance comparisons between multi-party objects. After that, every party encrypts its dataset objects along with the objects' μ-score using pk and sends it to the EDRS. Every party also sends the list of its $(k-1)$-skyband objects to the EDRS in plaintext. The remaining part of this chapter is described considering $K = k-1$, *i.e.*, K-skyband (KSB) denotes the $(k-1)$-skyband.

• **Computation by the EDRS and the EKSP:** The EDRS singly cannot compute the multi-party top-k dominating query from the encrypted objects of individual parties. Therefore, the EDRS and the EKSP collaboratively compute the multi-party top-k dominating query. However, during the secure computation, they reliably keep the dataset objects and the top-k query results hidden and protected from each other. In this regard, the EDRS and the EKSP compute through two intra-dependent protocols: the Secure top-k Dominating Query ($SKDQ$) protocol and the Fast Secure Dominance Comparison ($FSDC$) protocol. Here the $FSDC$ protocol compares the dominance relationship between two encrypted objects belong to different parties. And the $SKDQ$ protocol utilizes the encrypted results of the $FSDC$ protocol and securely computes the top-k dominating objects from the combined multi-party datasets. Here subsection 5.4.1 describes the $FSDC$ protocol, then subsection 5.4.2 the $SKDQ$ protocol.

5.4.1 Fast Secure Dominance Comparison (FSDC) Protocol

The $FSDC$ protocol securely compare the dominance relationship between two encrypted objects. The detailed construction of the $FSDC$ protocol is described in Algorithm 4.

At first, the EDRS applies homomorphic addition and multiplication properties of Paillier cryptosystem to expand D-dimensional encrypted objects $[\boldsymbol{P}(p_1, \cdots, p_D)]$ and $[\boldsymbol{Q}(q_1, \cdots, q_D)]$ into four $2D$ length encrypted vectors $[\boldsymbol{X}]$, $[\boldsymbol{X'}]$, $[\boldsymbol{Y}]$, and $[\boldsymbol{Y'}]$. The elements of these vectors can be given by the following equation: (i) $[x_i] := 2\hat{\times}[p_i]\hat{+}[1]$; (ii) $[x_{D+i}] := (-1)\hat{\times}[x_i]$; (iii) $[x'_i] := 2\hat{\times}[p_i]$; (iv) $\left[x'_{D+i}\right] := (-1)\hat{\times}[x'_i]$; (v) $[y_i] := 2\hat{\times}[q_i]$; (vi) $[y_{D+i}] := (-1)\hat{\times}[y_i]$; (vii) $[y'_i] := 2\hat{\times}[q_i]\hat{+}[1]$; (viii) $\left[y'_{D+i}\right] := (-1)\hat{\times}[y'_i]$;

Algorithm 4 Fast Secure Dominance Comparison (FSDC) Protocol

```
Input: EDRS has $[\boldsymbol{P}]$ and $[\boldsymbol{Q}]$; EKSP has private decryption key $sk$.
Output: EDRS obtains $[dom_P]$ and $[dom_Q]$.
EDRS:
Expands $[\boldsymbol{P}]$ and $[\boldsymbol{Q}]$ into four $2D$ length vectors $[\boldsymbol{X}]$, $[\boldsymbol{X'}]$, $[\boldsymbol{Y}]$ and $[\boldsymbol{Y'}]$;
Constructs two $2D$ length binary vectors $\boldsymbol{v} = (1_1, ..., 1_D, 0_{D+1}, ..., 0_{2D})$ and $\boldsymbol{v'} = (1_1, ..., 1_d, 0_{D+1}, ..., 0_{2D})$;
Generates two $2D$ length random binary vectors $\boldsymbol{s} = (s_1, ..., s_{2D})_{s_i \in 0,1}$ and $\boldsymbol{s'} = (s'_1, ..., s'_{2D})_{s'_i \in 0,1}$.
Calculates $[\boldsymbol{\delta}] = ([\delta_1], ..., [\delta_{2D}])$ and $[\boldsymbol{\delta'}] = ([\delta'_1], ..., [\delta'_{2D}])$;
    i. if $s_i = 0$ then $[\delta_i] := m \hat{\times} \left([x_i] \hat{-} [y_i]\right) \hat{+} [r]$ else $[\delta_i] := m \hat{\times} \left([y_i] \hat{-} [x_i]\right) \hat{+} [r]$;
    ii. if $s'_i = 0$ then $[\delta'_i] := m' \hat{\times} \left([y'_i] \hat{-} [x'_i]\right) \hat{+} [r']$ else $[\delta'_i] := m' \hat{\times} \left([x'_i] \hat{-} [y'_i]\right) \hat{+} [r']$;
    where $\{r, m, r', m'\} \in \mathbb{Z}^+$ such that $r < m$ and $r' < m'$
Computes $\boldsymbol{w} := \boldsymbol{v} \bigoplus \boldsymbol{s}$ and $\boldsymbol{w'} := \boldsymbol{v'} \bigoplus \boldsymbol{s'}$;
Uses random shuffling functions $\pi$ and $\pi'$ to shuffle vectors $[\boldsymbol{\delta}]$, $[\boldsymbol{\delta'}]$, $\boldsymbol{w}$ and $\boldsymbol{w'}$ to obtain vectors $[\boldsymbol{\theta}]$, $[\boldsymbol{\theta'}]$, $\boldsymbol{g}$ and $\boldsymbol{g'}$, respectively;
Uses hash function to compute $h \leftarrow H(\boldsymbol{g})$ and $h' \leftarrow H(\boldsymbol{g'})$;
Sends $[\boldsymbol{\theta}]$, $[\boldsymbol{\theta'}]$, $h$ and $h'$ to EKSP;
EKSP:
Decrypts $[\boldsymbol{\theta}]$ and $[\boldsymbol{\theta'}]$ using private decryption key $sk$;
Constructs two $2D$ length binary vectors $\boldsymbol{u} = (u_1, ..., u_{2D})$ and $\boldsymbol{u'} = (u'_1, ..., u'_{2D})$;
    a. if $\theta_i > N/2$ then $u_i := 0$ else $u_i := 1$;                              ▷ $N \in pk$
    b. if $\theta'_i > N/2$ then $u'_i := 0$ else $u'_i := 1$;
if $H(\boldsymbol{u}) = h$ and $H(\boldsymbol{u'}) \neq h'$ then $dom_P := 0$, $dom_Q := 1$;       ▷ $\boldsymbol{Q} \prec \boldsymbol{P}$
else if $H(\boldsymbol{u}) \neq h$ and $H(\boldsymbol{u'}) = h'$ then $dom_P := 1$, $dom_Q := 0$;  ▷ $\boldsymbol{P} \prec \boldsymbol{Q}$
else $dom_P := 0$, $dom_Q := 0$;                    ▷ $\boldsymbol{P}$ and $\boldsymbol{Q}$ do not dominate each other
end if
Sends $[dom_P]$ and $[dom_Q]$ to EDRS;
```

After expansion, the dominance relationship between two encrypted objects $[\boldsymbol{P}]$ and $[\boldsymbol{Q}]$ will be turned to two vector comparison problems: i. compare $[\boldsymbol{X}]$ and $[\boldsymbol{Y}]$, and ii. compare $[\boldsymbol{X'}]$ and $[\boldsymbol{Y'}]$. The EDRS also constructs two $2D$ length binary vectors $\boldsymbol{v}$ and $\boldsymbol{v'}$ to mark the expected comparison result between $[\boldsymbol{X}]$ and $[\boldsymbol{Y}]$, and between $[\boldsymbol{X'}]$ and $[\boldsymbol{Y'}]$. Particularly, if $v_i = 1$, the EDRS expects $x_i > y_i$ in position i; and if $v_i = 0$, the EDRS expects $x_i < y_i$. In contrast, $v'_i = 1$ represent the EDRS's expectation of $x'_i < y'_i$ in position i and vice versa.

Next, the EDRS generates two $2D$ length random binary vectors $\boldsymbol{s}$ and $\boldsymbol{s'}$. Then calculates $[\boldsymbol{\delta}]$ and $[\boldsymbol{\delta'}]$ according to Step 5 of Algorithm 4. The EDRS also computes $\boldsymbol{w} := \boldsymbol{v} \bigoplus \boldsymbol{s}$ and $\boldsymbol{w'} := \boldsymbol{v'} \bigoplus \boldsymbol{s'}$, since $\boldsymbol{s}$ and $\boldsymbol{s'}$ swap the expected comparison result.

After that, the EDRS generates random shuffling function π and π' to compute $[\boldsymbol{\theta}] \leftarrow \pi([\boldsymbol{\delta}])$, $\boldsymbol{g} \leftarrow \pi(\boldsymbol{w})$, $[\boldsymbol{\theta'}] \leftarrow \pi([\boldsymbol{\delta'}])$, and $\boldsymbol{g'} \leftarrow \pi(\boldsymbol{w'})$. After shuffling, the EDRS uses hash function to compute the hash value of binary vectors $\boldsymbol{g}$ and $\boldsymbol{g'}$ *i.e.* $h \leftarrow H(\boldsymbol{g})$ and $h' \leftarrow H(\boldsymbol{g'})$. Then, the EDRS sends $[\boldsymbol{\theta}]$, $[\boldsymbol{\theta'}]$, h and h' to EKSP.

After receiving the encrypted vectors along with the expected hash results, the EKSP decrypts $[\boldsymbol{\theta}]$ and $[\boldsymbol{\theta'}]$ using its private decryption key sk. From the decrypted vectors, the EKSP constructs two $2D$ length binary vectors $\boldsymbol{u}$ and $\boldsymbol{u'}$ according to Step 12 of Algorithm 4. Then by comparing the hash values of $\boldsymbol{u}$ with h and $\boldsymbol{u'}$ with h', the EKSP computes the dominance relation between $\boldsymbol{P}$ and $\boldsymbol{Q}$ according to Step 13 to Step 16 of Algorithm 4. At last, the EKSP sends the encrypted values of dom_S and dom_T to the EDRS.

5.4.2 Secure top-k Dominating Query (SKDQ) Protocol

The *SKDQ* protocol is designed for the computation of the multi-party top-k dominating objects from the encrypted multi-party datasets. This protocol restricts that only the respective party could be able to recognize its multi-party top-k dominating objects.

Algorithm 5 Secure top-k Dominating Query ($\boldsymbol{SKDQ}$) Protocol

```
Input: EDRS has encrypted objects set $[\boldsymbol{DS}_l]$ of $\boldsymbol{Party_l}$ $(l = 1, 2, \cdots, L)$ along with the $[\mu_{l,i}]$ corresponding to $\boldsymbol{O}_{l,i} \in \boldsymbol{DS}_l$ and the list of objects belongs to $KSB(\boldsymbol{DS}_l)$, EKSP has private decryption key $sk$;    $\triangleright$ $K = k - 1$
Output: Each participating party $\boldsymbol{Party_l}$ $(l = 1, 2, \cdots, L)$ recognizes its multi-party top-$k$ dominating objects;
for all pair of $\{(\boldsymbol{O}_{a,i}, \boldsymbol{O}_{b,j}) \,|\, \boldsymbol{O}_{a,i} \in KSB(\boldsymbol{DS_a}),\ \boldsymbol{O}_{b,j} \in \boldsymbol{DS_b},\ a \neq b\}$ do
    EDRS and EKSP:
    Compute $\left([dom_{\boldsymbol{O}_{a,i}}], [dom_{\boldsymbol{O}_{b,j}}]\right) \leftarrow \boldsymbol{FSDC}([\boldsymbol{O}_{a,i}], [\boldsymbol{O}_{b,j}])$;
    EDRS:
    Computes (i) $[\mu_{a,i}] := [\mu_{a,i}] \hat{+} [dom_{\boldsymbol{O}_{a,i}}]$ and (ii) $[\mu_{b,j}] := [\mu_{b,j}] \hat{+} [dom_{\boldsymbol{O}_{b,j}}]$;
end for
for all $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB(\boldsymbol{DS}_l)$ do
    EDRS:
    Computes $[\mu'_{l,i}] := [\boldsymbol{S}] \hat{-} [\mu_{l,i}]$ where $\boldsymbol{S} = \sum_{l=1}^{L} |\boldsymbol{DS}_l|$;
    Computes $[c_{l,i}] := \left(2^{\epsilon} \hat{\times} [\mu'_{l,i}]\right) \hat{+} [z]$, where $\epsilon := \lceil \log_2 \left(\sum_{l=1}^{L} |KSB(\boldsymbol{DS}_l)|\right) \rceil$ and $z$ is an exclusive integer between 0 to $2^{\epsilon}$;
    Initializes $[t_{l,i}] := [1]$;
end for
for $j = 1$ to $k$ do
    EDRS and EKSP:
    Compute $[min] \leftarrow SMIN_n($ all $[c_{l,i}]$ corresponding to $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB(\boldsymbol{DS}_l))$;
    EDRS:
    for all $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB(\boldsymbol{DS}_l)$ do
        Computes $[c_{l,i}] := [c_{l,i}] \hat{-} [min]$;
        Computes $[\lambda_{l,i}] := r \hat{\times} [c_{l,i}]$, where $r \in \mathbb{Z}^{>\mathbf{1}}$;
    end for
    Applies random shuffling function $\Pi$ to obtain $[\boldsymbol{\omega}] \leftarrow \Pi([\boldsymbol{\lambda}])$;
    Sends $[\boldsymbol{\omega}]$ to EKSP;
    EKSP:
    for all $\omega_i \in \boldsymbol{\omega}$, if $\omega_i = 0$ set $\omega'_i := 1$, else set $\omega'_i := 0$;
    Sends $[\boldsymbol{\omega'}]$ to EDRS;
    EDRS:
    Applies inverse function $\Pi^{-1}$ to obtain $[\boldsymbol{\lambda'}] \leftarrow \Pi^{-1}([\boldsymbol{\omega'}])$;
    if $j < k$ then
        for all $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB\left(\boldsymbol{DS}_l\right)$ do
            EDRS and EKSP:
            Compute $\left[t_{l,i}\right] \leftarrow SM\left(\left[t_{l,i}\right], [1] \hat{-} \left[\lambda'_{l,i}\right]\right)$;
            EDRS:
            Computes $\left[c_{l,i}\right] := \left[c_{l,i}\right] \hat{+} \left(\left(2^{\epsilon} \times \boldsymbol{S}\right) \hat{\times} \left[\lambda'_{l,i}\right]\right)$;
        end for
    else
        EDRS:
        Initializes $\left[\mu_k\right] := [0]$;
        for all $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB\left(\boldsymbol{DS}_l\right)$ do
            EDRS and EKSP:
            Compute $[temp] \leftarrow SM\left(\left[\mu_{l,i}\right], \left[\lambda'_{l,i}\right]\right)$;
            EDRS:
            Computes $\left[\mu_k\right] := \left[\mu_k\right] \hat{+} [temp]$;
        end for
        for all $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB\left(\boldsymbol{DS}_l\right)$ do
            EDRS:
            Computes $\left[\mu_{l,i}\right] := \left[\mu_k\right] \hat{-} \left[\mu_{l,i}\right]$;
            EDRS and EKSP:
            Compute $\left[t_{l,i}\right] \leftarrow SM\left(\left[t_{l,i}\right], \left[\mu_{l,i}\right]\right)$;
        end for
    end if
end for
EDRS:
for each $\boldsymbol{O}_{l,i} \in \bigcup_{l=1}^{L} KSB\left(\boldsymbol{DS}_l\right)$, computes $\left[t'_{l,i}\right] := \left(r \hat{\times} \left[t_{l,i}\right]\right) \hat{+} \left[\gamma_{l,i}\right]$, where $r \in \mathbb{Z}^{>1}$ and $\gamma_{l,i} \in \mathbb{Z}^{+}$;
Sends the array $[\boldsymbol{t'}]$ to EKSP and each $\gamma_{l,i}$ to corresponding $\boldsymbol{Party}_l$;
EKSP:
Decrypts $[\boldsymbol{t'}]$ and sends each $t'_{l,i}$ to corresponding $\boldsymbol{Party}_l$;
Each $\boldsymbol{Party}_l$ $(l = 1, \cdots, L)$:
Computes $tk'_{l,i} := t'_{l,i} - \gamma_{l,i}$, and recognizes $\boldsymbol{O}_{l,i}$ as its multi-party top-$k$ dominating object if $tk'_{l,i} = 0$;
```

Since the top-k dominating objects belong to K-skyband objects, this protocol only focuses on computing encrypted μ-score of every K-skyband object of all parties' datasets. Algorithm 5 describes the $SKDQ$ protocol.

The $SKDQ$ protocol utilizes the $FSDC$ protocol to compare the dominance relation securely between every object $\boldsymbol{O}_{a,i} \in \bigcup\limits_{a=1}^{L} KSB\left(\boldsymbol{DS}_a\right)$ with another object $\boldsymbol{O}_{b,j} \in \bigcup\limits_{b=1}^{L} \boldsymbol{DS}_b$, such that, $a \neq b$. The encrypted result of the $FSDC$ protocol is used for computing the encrypted multi-party μ-score $[\mu_{l,i}]$ of object $\boldsymbol{O}_{l,i} \in \bigcup\limits_{l=1}^{L} KSB\left(\boldsymbol{DS}_l\right)$.

Since the EDRS only obtains the μ-score of every K-skyband object in ciphertext and the framework does not reveal the computation result to both the EDRS and the EKSP, Step 7 to Step 58 of Algorithm 5 has been designed for recognizing the top-K dominating objects by the individual parties.

For every K-skyband objects of the participating parties, the EDRS computes $\left[\mu'_{l,i}\right] := [\boldsymbol{S}] \hat{-} [\mu_{l,i}]$ according to Step 9 of Algorithm 5. Here $\boldsymbol{S}$ denotes the total number of objects within the combined multi-party datasets. Since the array $\boldsymbol{\mu'}$ is obtained through a linearly decreasing function over the array $\boldsymbol{\mu}$, finding the top-k objects with the highest values within the encrypted array $[\boldsymbol{\mu}]$ is equivalent to finding the top-k objects with the lowest values within the encrypted array $[\boldsymbol{\mu'}]$.

A potential issue may occur when multiple elements of array $[\boldsymbol{\mu'}]$ have the same minimum value. To avoid such occurrences of multiple minimum values, the EDRS computes $[c_{l,i}]$ for each $\left[\mu'_{l,i}\right] \in [\boldsymbol{\mu'}]$ according to Step 10 of Algorithm 5. The values within the encrypted array $[\boldsymbol{c}]$ will be unique, while their sorting order will be equivalent to the values within encrypted array $[\boldsymbol{\mu'}]$. Besides, to mark the multi-party top-k dominating objects, the EDRS also creates an encrypted array $[\boldsymbol{t}]$ for the K-skyband objects of all parties.

For the k number of cycles, the EDRS and the EKSP compute through Step 14 to Step 50 of Algorithm 5. Within each cycle, the EDRS and the EKSP securely compute the minimum encrypted value $[min]$ from the encrypted elements of array $[\boldsymbol{c}]$. Then, the

EDRS subtracts $[min]$ from each element of $[\boldsymbol{c}]$ using additive homomorphism property of the Paillier cryptosystem. For masking the array $[\boldsymbol{c}]$, the EDRS computes $[\lambda_{l,i}]$ according to Step 19 of Algorithm 5. Note that, each $\lambda_{l,i}$ is an uniformly random positive value except when $c_{l,i} - min = 0$, in which case $\lambda_{l,i} = 0$. Then the EDRS shuffles the elements of encrypted array $[\boldsymbol{\lambda}]$ using random shuffling function Π and sends the shuffled array $[\boldsymbol{\omega}]$ to the EKSP. The EKSP computes $\boldsymbol{\omega'}$ according to Step 24 of Algorithm 5 and sends the encrypted array $[\boldsymbol{\omega'}]$ to the EDRS. After that, the EDRS obtains $[\boldsymbol{\lambda'}]$ by applying inverse function Π^{-1} on $[\boldsymbol{\omega'}]$.

For each cycle, only one element of the encrypted array $[\boldsymbol{\lambda}]$ will be 1. Therefore, every cycle between 1 upto $(k-1)$, multiplying $\left[1 - \lambda'_{l,i}\right]$ with $[t_{l,i}]$ using SM protocol only alters a single element of the encrypted array $[\boldsymbol{t}]$ into a encrypted value of zero. Furthermore, the EDRS securely adds a large value with $[c_{l,i}]$, in which case $\lambda'_{l,i} = 1$. This way, the corresponding $[c_{l,i}]$ will not be the minimum value in the remaining iterations.

Noted that, the objects, whose μ-score is equal to the μ-score of the k^{th} rank dominating object, are also top-k dominating objects. Therefore, in the k^{th} cycle, the EDRS and the EKSP securely mark all the objects whose μ-scores are equal to the μ-score of the k^{th} rank dominating object. In this regard, at first, the EDRS and the EKSP applies Step 36 to Step 43 of Algorithm 5 to obtain $[\mu_k]$ as the encrypted μ-score of the k^{th} rank dominating object. Then, by using Step 44 to Step 49 of Algorithm 5, they securely change every encrypted value $[t_{l,i}]$ into $[0]$ if $\boldsymbol{O}_{l,i}$ is a k^{th} rank dominating object..

At last, EDRS computes $\left[t'_{l,i}\right]$ according to Step 53, and sends the array $[\boldsymbol{t'}]$ to EKSP and each random positive integer $\gamma_{l,i}$ to $\boldsymbol{Party}_l$. Then the EKSP decrypts $[\boldsymbol{t'}]$ and sends each $t'_{l,i}$ to $\boldsymbol{Party}_l$. Finally, every $\boldsymbol{Party}_l$ recognizes its multi-party top-k dominating objects using Step 58 of Algorithm 5.

5.5 Privacy and Security Analyses

The privacy and security achieved by this framework is explained here.

• **Privacy of *FSDC* protocol:** Due to the encryption of $[P]$ and $[Q]$, the EDRS cannot obtain the plaintext value of objects $\boldsymbol{P}$ and $\boldsymbol{Q}$. In contrast, the EDRS also prevents the EKSP to know the objects $\boldsymbol{P}$ and $\boldsymbol{Q}$ by using random transformation and permutation processes. Since the EDRS generates random binary vector $\boldsymbol{s}$ and $\boldsymbol{s'}$ and utilizes them during the computation of $[\boldsymbol{\delta}]$ and $[\boldsymbol{\delta'}]$, the EKSP cannot know whether the encrypted value of $[\delta_i]$ is $x_i - y_i$ or $y_i - x_i$, and the value of $[\delta'_i]$ is $x'_i - y'_i$ or $y'_i - x'_i$. Also, since the EDRS masks $[\delta_i]$ and $[\delta'_i]$ with random positive integers $\{r, m, r', m'\}$, the EKSP cannot obtain the exact value of $|x_i - y_i|$ and $|x'_i - y'_i|$. Furthermore, the EKSP cannot know the comparison result for some specific attribute because the EDRS shuffle the vectors $[\boldsymbol{\delta}]$ and $[\boldsymbol{\delta'}]$ using the random shuffling functions π and π'. On the other hand, due to the encryption of the dominance comparison result by the EKSP, the EDRS cannot know the dominance relation between $\boldsymbol{P}$ and $\boldsymbol{Q}$.

• **Privacy of *SKDQ* protocol:** Since the EDRS only achieves the encrypted dataset objects of the participating parties, it cannot know the original dataset objects. Furthermore, within Algorithm 5 it can be observed that, during the computation, the EDRS obtains the objects' μ-score and all the intermediate results only in ciphertext. So, the EDRS cannot identify the top-k dominating objects with the highest μ-score. On the other hand, the EKSP can only achieve the randomly masked values and randomly shuffled array of encrypted values during the computation through the $SKDQ$ protocol. Hence the EKSP also cannot know anything from the secure computation process. Therefore, it can be claimed that the proposed framework maintains the privacy of the participating parties' objects and the query results during the computation of multi-party top-k dominating query.

5.6 Performance Evaluation

This section evaluates the performance and effectiveness of the proposed framework. Noted that the computation of every individual party is identical either for the non-secured distributed top-k dominating query or for the privacy-preserving multi-party top-k dominating query. Therefore, the author only examined the runtime of the proposed privacy-preserving multi-party top-k dominating query ($SKDQ$ protocol).

The author used two identical computers connected with Cisco Catalyst 2960-X Series Gigabit Switch for the experimental setup. Out of these, one computer was considered as the EKSP and another one as the EDRS. Each of these computers has an Intel® Core i5-6500 3.20GHz CPU and 8GB memory. The author used the 64-bit Ubuntu 16.04 operating system for the experiment and built the program using Java RMI, where the EKSP holds the remote method with the private key, and the EDRS has the client method with the public key. The author generated two synthetic datasets to simulate the datasets of two participating parties. Each attribute value of the synthetic datasets was randomly picked from 16-bit unsigned integer. From the complexity analysis of the proposed framework, it is noticeable that there are four factors can influence the total running time of the $SKDQ$ protocol and the $FSDC$ protocol. These are the value of k, the data distribution, the dimension of the data objects, and the number of dataset objects. The 96-bit Paillier cryptosystem key was used for the experimental setup.

• **Effect of data distribution:** The author used two fixed-size datasets (500 two-dimensional objects per dataset) with independent, correlated, and anti-correlated data distributions to study the effect of data distribution over the proposed framework. Fig. 5.2 shows the effect of data distribution while varying k from 2 to 10. Since the number of K-skyband objects within each party's dataset varies with the data distribution, the runtime of the $SKDQ$ protocol is also effected by data distribution. The simulation results also reflect it.

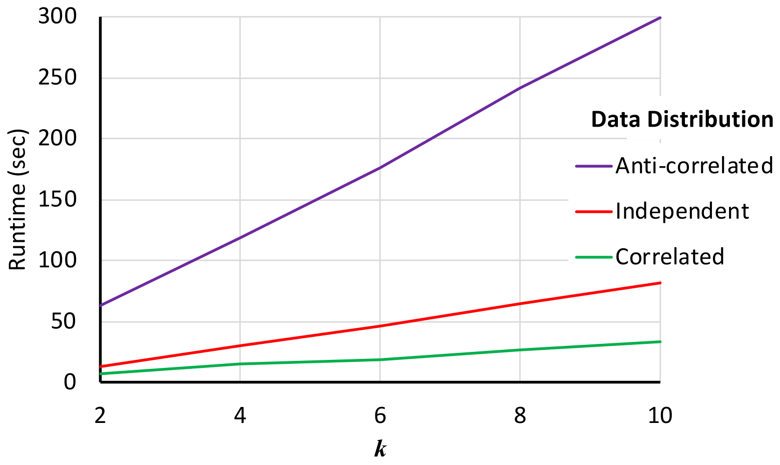


Figure 5.2: Runtime of the multi-party top-k dominating query for various data distribution [Object dimension: 2, No. of objects: 500 objects/dataset]

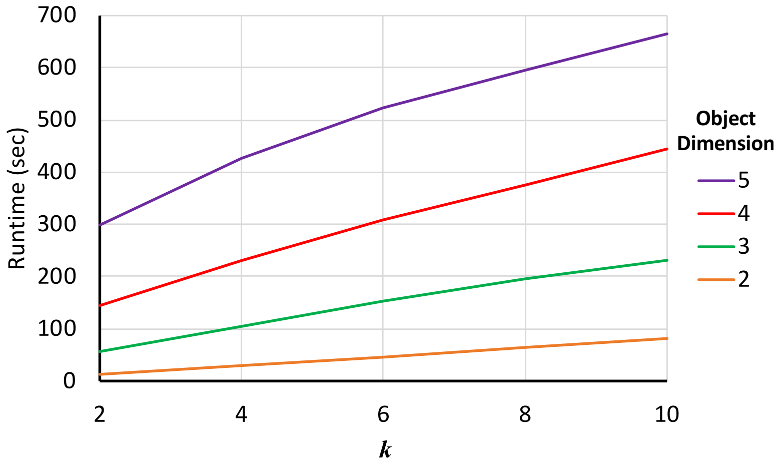


Figure 5.3: Runtime of the multi-party top-k dominating query for varied object dimension [Data distribution: Independent, No. of objects: 500 objects/dataset]

• **Effect of object dimension:** To observe the effect of the object dimension, The author used two fixed-size datasets (500 objects per dataset) with varied object dimensions. Fig. 5.3 shows the runtime comparison of the $SKDQ$ protocol with respect to varied object dimension, and varied k. It is apparent that the computation complexity of the FSDC protocol varies based on the object dimension. Moreover, the object dimension also affects the number of k-skyband objects within each partys dataset. Therefore, the object dimension also influences the runtime of the $SKDQ$ protocol.

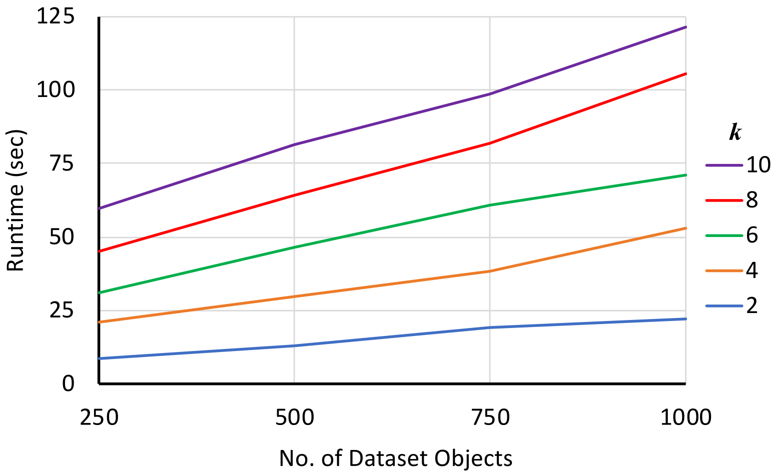


Figure 5.4: Runtime of the multi-party top-k dominating query for varied number of objects [Data distribution: Independent, Object dimension: 2]

• **Effect of dataset size:** To analyze the effect of dataset size, the author used two 2-dimensional independently distributed datasets. One of the datasets used fixed 500 data objects, and the data objects in the other dataset varied from 250 to 1000. Fig. 5.4 illustrates the runtime of the $SKDQ$ protocol for varied number of objects within the dataset, and varied k.

5.7 Concluding Remarks

In this chapter, the author presents another novel framework for recognizing multi-party top-k dominating objects where any party does not need to expose its dataset to others. The author describes that the framework also maintains the privacy of the query results. Although the designed $FSDC$ protocol utilizes anonymization schemes, it does not lose the generality of the dominance relation. Thus, the proposed framework can yield the correct result. Moreover, the prior computation of $k-1$-skyband and μ-score by the participating parties in plaintext improve the efficiency of the proposed framework.

Chapter 6

Conclusion

In this chapter, at first, Section 6.1 discusses some applications of the proposed frameworks. Then Section 6.2 discusses the key contribution of the author in this dissertation. And finally, Section 6.3 discusses the future research scope to extend this work.

6.1 Applications of proposed models

The framework proposed in Chapter 3 for the privacy-preserving multi-party skyline query can be applied in such a situation, where every party wants to identify their multi-party skyline objects without revealing their dataset to others. The framework also considers the privacy of the dominance comparison results between the multi-party objects. Without incorporating any third party, the framework can achieve the desired computation goal and also can meet the privacy requirements. This framework creates opportunities for the organizations to identify their multi-party skyline objects from their datasets when all organizations do not trust any common third party(s) together.

Chapter 4 discusses a privacy-preserving K-skyband computation model in the union of multi-party databases, which computes the K-skyband without disclosing the objects in any party's dataset. The author introduces an efficient way to transform the multi-party objects' attributes without altering their sorting order rank and utilizes the transformed sorting order rank of the objects' attributes for computing multi-party K-skyband. Since the proposed scheme does not apply the secure dominance comparison of multi-party objects, it can be considered as the most efficient approach to process the skyline query and its variants.

The framework introduced in Chapter 5 extracts top-k dominating objects from the union of multi-party databases in a privacy-preserving way. This framework can locate top-k objects from the combined databases of multiple parties, while it is difficult to define a scoring function for the selection process. It selects the top-k objects based on the 'domination score'. This model can be applied in such circumstances when multiple parties want to identify their most important sensitive database objects, which dominate the maximum number of objects from their combined databases.

6.2 Contributions

In recent years, the skyline query and its variants are known to be popular queries for selecting representative objects from a large dataset, namely the 'big data'. Similarly, the distributed data mining approach has been extensively applied to many applications. Meanwhile, protecting the privacy and security of sensitive data during data processing has become a research hotspot. This dissertation has extensively studied the privacy and security issues in the distributed processing of the skyline query and its variants.

Contribution on Problem I: Chapter 3 introduces a novel framework for the privacy-preserving multi-party skyline query. The framework ensures the privacy of every participating party's objects and also maintained the privacy of the individual dominance comparison results between multi-party objects. The framework does not require any trusted third party. Thus, it always keeps the privacy of the honest parties' objects even if an adversary compromise one or more dishonest parties. Chapter 3 presents detailed algorithms and the data-flow diagrams of the underlying protocols of the proposed framework. Besides, the privacy and security analyses, complexity analyses, and performance evaluation of the proposed framework are also explained.

Contribution on Problem II: An efficient solution for the privacy-preserving multi-party K-skyband query is described in Chapter 4. According to [7], the dominance relation between the objects can also be computed from the sorting order rank of the objects' attributes. The proposed solution utilizes this vital property of the dominance relation and offers an efficient scheme to transform the multi-party objects' attributes without altering their sorting order rank. Then a trusted third party computes the multi-party K-skyband from the order rank unchanged transformed values of the objects' attributes. Chapter 4 also presents the performance evaluation of the proposed solution through the experimental simulation of the proposed system model.

Contribution on Problem III: A framework for privacy-preserving multi-party top-k dominating query is presented in Chapter 5. The author introduces a cloud-based novel way for computing the domination score (μ-score) of the encrypted multi-party dataset objects by incorporating two cloud-based semi-honest computation service providers. The author ensures the privacy and security of the encrypted multi-party objects while at least one of the cloud service providers is honest and does not compromise by adversaries or dishonest parties. The author shows that both cloud service providers cannot know anything about the participating parties' objects as well as the query result through the proposed framework. In addition, the author also analyzes the privacy and security of the proposed model and shows the performance through experimental simulation.

6.3 Future Direction

The framework for the privacy-preserving multi-party skyline query (Chapter 3) is more realistic since it does not incorporate any semi-honest third party. However, it is not as efficient as the multi-party K-skyband query framework proposed in Chapter 4. Its efficiency can be improved by adapting the parallelly distributed computation environment, *e.g.*, MapReduce.

The privacy-preserving multi-party K-skyband query framework (Chapter 4) can also be applied for computing other variants of the skyline query such as K-dominant skyline and top-k dominating query. Besides, there exist some query operations, which can be computed by applying the query to the rank of the objects' attributes, *e.g.*, count aggregate function. The proposed multi-party objects' attributes transformation scheme can also be applied efficiently and securely to such query operations in the union of sensitive multi-party databases.

Furthermore, a study is necessary to design optimization mechanisms for the privacy-preserving multi-party top-K query dominating query (Chapter 5). This framework can also be extended to rank multi-party objects in a privacy-preserving way.

The spatial skyline query in a privacy-preserving way can also be a novel research area to expand this work. Such computation can benefit both query users and the data owner in such circumstances where the users do not reveal the queries to the database owner, and the data owner also does not disclose the entire database to the users except the query results. Besides, privacy-preserving social media data mining will become an emerging research field in the near future.

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 439–450, New York, NY, USA, 2000. ACM.

[2] D. Amagata, Y. Sasaki, T. Hara, and S. Nishio. Efficient processing of top-k dominating queries in distributed environments. *World Wide Web*, 19(4):545–577, Jul 2016.

[3] W.-T. Balke, U. Güntzer, and J. X. Zheng. Efficient distributed skylining for web information systems. In *Proceedings of the 9th International Conference on Extending Database Technology (EDBT)*, pages 256–273, Heraklion, Crete, Greece, 2004. Springer Berlin Heidelberg.

[4] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Proceedings 17th International Conference on Data Engineering*, pages 421–430, April 2001.

[5] G. Chen and Y. Wang. Top-k dominating query processing over distributed data streams. *Global Journal of Engineering Science and Research Management, 5(6),*, page 1323, 2018.

[6] L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu. isky: Efficient and progressive skyline computing in a structured p2p network. In *2008 The 28th International Conference on Distributed Computing Systems*, pages 160–167, June 2008.

[7] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In *Proceedings 19th International Conference on Data Engineering*, pages 717–719, March 2003.

[8] Y. Elmehdwi, B. K. Samanthula, and W. Jiang. Secure k-nearest neighbor query over encrypted data in outsourced environments. In *2014 IEEE 30th International Conference on Data Engineering*, pages 664–675, March 2014.

[9] P. Ezatpoor, J. Zhan, J. M. Wu, and C. Chiu. Finding top- k dominance on incomplete big data using mapreduce framework. *IEEE Access*, 6:7872–7887, 2018.

[10] Y. Gao, X. Miao, H. Cui, G. Chen, and Q. Li. Processing k-skyband, constrained skyline, and group-by skyline queries on incomplete data. *Expert Systems with Applications*, 41(10):4959 – 4974, 2014.

[11] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[12] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, NY, USA, 1987. ACM.

[13] Z. Gong, G.-Z. Sun, J. Yuan, and Y. Zhong. Efficient top-k query algorithms using k-skyband partition. In Peter Mueller, Jian-Nong Cao, and Cho-Li Wang, editors, *Scalable Information Systems*, pages 288–305, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[14] K. Hamada, D. Ikarashi, K. Chida, and K. Takahashi. Oblivious radix sort: An efficient sorting algorithm for practical secure multi-party computation. *IACR Cryptology ePrint Archive*, 2014:121, 2014.

[15] K. Hose and A. Vlachou. A survey of skyline processing in highly distributed environments. *The International Journal on Very Large Data Bases*, 21(3):359–384, Jun 2012.

[16] 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 of Things Journal*, pages 1–1, 2018.

[17] F. Kerschbaum, D. Biswas, and S. d. Hoogh. Performance comparison of secure comparison protocols. In *2009 20th International Workshop on Database and Expert Systems Application*, pages 133–136, Aug 2009.

[18] M. Kontaki, A. N. Papadopoulos, and Y. Manolopoulos. Continuous top-k dominating queries. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):840–853, May 2012.

[19] D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*, pages 275–286. VLDB Endowment, 2002.

[20] C. Li, Annisa, A. Zaman, and Y. Morimoto. Mapreduce-based computation of area skyline query for selecting good locations in a map. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4779–4782, Dec 2017.

[21] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76 – 85, 2017.

[22] X. Lian and L. Chen. Probabilistic top-k dominating queries in uncertain databases. *Information Sciences*, 226:23 – 46, 2013.

[23] H.-Y. Lin and W.-G. Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *Applied Cryptography and Network Security*, pages 456–466, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[24] Z. Lin and J. W. Jaromczyk. An efficient secure comparison protocol. In *2012 IEEE International Conference on Intelligence and Security Informatics*, pages 30–35, June 2012.

[25] Y. Lindell and B. Pinkas. Privacy preserving data mining. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, pages 36–54, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[26] J. Liu, J. Yang, L. Xiong, and J. Pei. Secure and efficient skyline queries on encrypted data. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2018.

[27] X. Liu, R. Deng, K. R. Choo, and Y. Yang. Privacy-preserving outsourced clinical decision support system in the cloud. *IEEE Transactions on Services Computing*, pages 1–1, 2017.

[28] X. Liu, S. Li, X. Chen, G. Xu, X. Zhang, and Y. Zhou. Efficient solutions to two-party and multiparty millionaires problem. *Security and Communication Networks*, 2017(5207386):11, 2017.

[29] X. Liu, R. Lu, J. Ma, L. Chen, and H. Bao. Efficient and privacy-preserving skyline computation framework across domains. *Future Generation Computer Systems*, 62:161 – 174, 2016.

[30] K. Mullesgaard, J. L. Pedersen, H. Lu, and Y. Zhou. Efficient skyline computation in mapreduce. In *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*, pages 37–48, Athens, Greece, 2014.

[31] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Proceedings of Advances in Cryptology - Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[32] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *ACM Transactions Database Systems*, 30(1):41–82, March 2005.

[33] Y. Rahulamathavan, R. C. . Phan, S. Veluru, K. Cumanan, and M. Rajarajan. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. *IEEE Transactions on Dependable and Secure Computing*, 11(5):467–479, Sep. 2014.

[34] J. B. Rocha-Junior, A. Vlachou, C. Doulkeridis, and K. Nørvåg. Agids: A grid-based strategy for distributed skyline query processing. In A. Hameurlain and A. M. Tjoa, editors, *Data Management in Grid and Peer-to-Peer Systems*, pages 12–23, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[35] H.-C. Ryu and S. Jung. Mapreduce-based skyline query processing scheme using adaptive two-level grids. *Cluster Computing*, 20(4):3605–3616, December 2017.

[36] B. K. K. Samanthula, H. Chun, and W. Jiang. An efficient and probabilistic secure bit-decomposition. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 541–546, New York, NY, USA, 2013. ACM.

[37] M. Sepehri, S. Cimato, and E. Damiani. Privacy-preserving query processing by multi-party computation. *The Computer Journal*, 58(10):2195–2212, October 2015.

[38] M. A. Siddique, H. Tian, and Y. Morimoto. Distributed skyline computation of vertically splitted databases by using mapreduce. In W.-S. Han, M. L. Lee, A. Muliantara, N. A. Sanjaya, B. Thalheim, and S. Zhou, editors, *Database Systems for Advanced Applications*, pages 33–45, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[39] T. Veugen, F. Blom, S. J. A. de Hoogh, and Z. Erkin. Secure comparison protocols in the semi-honest model. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1217–1228, Oct 2015.

[40] S. Wang, B. C. Ooi, A. K. H. Tung, and L. Xu. Efficient skyline query processing on peer-to-peer networks. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 1126–1135, April 2007.

[41] Wikipedia. Secure multi-party computation. In *https://en.wikipedia.org/wiki/Secure_multi-party_computation*. [online] Wikipedia, [Accessed 18 November, 2019].

[42] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[43] M. L. Yiu and N. Mamoulis. Efficient processing of top-k dominating queries on multidimensional data. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pages 483–494, University of Vienna, Austria, 2007.

[44] A. Zaman, M. A. Siddique, Annisa, and Y. Morimoto. Secure computation of skyline query in mapreduce. In J. Li, X. Li, S. Wang, J. Li, and Q. Z. Sheng, editors, *Advanced Data Mining and Applications*, pages 345–360, Cham, 2016. Springer International Publishing.

[45] W. Zhang, X. Lin, Y. Zhang, J. Pei, and W. Wang. Threshold-based probabilistic top-k dominating queries. *The International Journal on Very Large Data Bases*, 19(2):283–305, Apr 2010.

List of Referred Publications

Referred Journals

J-1 Mahboob Qaosar, Kazi Md. Rokibul Alam, Asif Zaman, Chen Li, Saleh Ahmed, Md. Anisuzzaman Siddique, Yasuhiko Morimoto. *“A Framework for Privacy-Preserving Multi-Party Skyline Query Based on Homomorphic Encryption”*, IEEE Access, ISSN 2169-3536, Vol: 7, Pages 167481-167496, doi: 10.1109/ACCESS.2019.2954156, November 2019.

J-2 Mahboob Qaosar, Asif Zaman, Md. Anisuzzaman Siddique, Chen Li, Yasuhiko Morimoto. *“Secure K-skyband computation framework in distributed multi-party databases”*, Information Sciences, ISSN 0020-0255, Vol: 515, Pages 388-403, doi: 10.1016/j.ins.2019.12.027, April 2020.

Referred International Conferences

C-1 Mahboob Qaosar, Kazi Md. Rokibul Alam, Chen Li, Yasuhiko Morimoto. *“Privacy-preserving Top-k Dominating Queries in Distributed Multi-party Databases”*, Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Pages 5794-5803, Los Angeles, CA, USA, December 9-12, 2019.

Other Publications (not in the dissertation)

Referred Journals

J-3 Mahboob Qaosar, Asif Zaman, Md. Anisuzzaman Siddique, Annisa, Yasuhiko Morimoto. *"Privacy-Preserving Secure Computation of Skyline Query in Distributed Multi-Party Databases"*, Information, MDPI, Switzerland, Vol: 10, Issue 3, Article No. 119(1-20), doi:10.3390/ info10030119, March 2019.

J-4 Md. Anisuzzaman Siddique, Hao Tian, Mahboob Qaosar, Yasuhiko Morimoto. *"MapReduce Algorithm for Variants of Skyline Queries: Skyband and Dominating Queries"*, Algorithms, MDPI, Switzerland, Vol: 12, Issue 8, Article No. 166(1-14), doi:10.3390/a12080166, August 2019.

J-5 Saleh Ahmed, Mahboob Qaosar, Asif Zaman, Md. Anisuzzaman Siddique, Chen Li, Kazi Md. Rokibul Alam, Yasuhiko Morimoto. *"Privacy-Aware MapReduce Based Multi-Party Secure Skyline Computation"*, Information, MDPI, Switzerland, Vol: 10, Issue 6, Article No. 207(1-19), doi:10.3390/info10060207, June 2019.

J-6 Chen Li, Annisa Annisa, Asif Zaman, Mahboob Qaosar, Saleh Ahmed, Yasuhiko Morimoto. *"MapReduce Algorithm for Location Recommendation by Using Area Skyline Query"*, Algorithms, MDPI, Switzerland, Vol: 11, Issue 12, Article No. 191(1-15), doi:10.3390/a11120191, November 2018.

Referred International Conferences

C-2 Mahboob Qaosar, Saleh Ahmed, Chen Li, Yasuhiko Morimoto. *"Hybrid Sensing and Wearable Smart Device for Health Monitoring and Medication: Opportunities and Challenges"*, The 2018 AAAI Spring Symposium Series Technical Report on Beyond Machine Intelligence: Understanding Cognitive Bias and Humanity for Well-Being AI, Pages 269-274, Palo Alto, California, USA, March 26-28, 2018.

C-3 Chen Li, Xu Zhang, Mahboob Qaosar, Saleh Ahmed, Kazi Md. Rokibul Alam, Yasuhiko Morimoto. *"Multi-factor Based Stock Price Prediction Using Hybrid Neural Networks with Attention Mechanism"*, 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/ CBDCom/CyberSciTech), Fukuoka, Japan, Pages 961-966, August 5-8, 2019.