A Framework for Multi-party Skyline Query Maintaining Privacy and Data-integrity

***Abstract*—Skyline query is well-known to find out the dominant objects from a large number of datasets. While multiple organizations want to analyze their combined dataset, skyline queries can assist in this regard. Maintaining privacy along with the data integrity of participating organizations' datasets is important because their commercial success depends on the result of these queries. This paper proposes a new framework for the multi-party skyline query that encompasses both privacy and data integrity. To ensure the privacy of participants' datasets, it adopts commutative encryptions by employing multiple independent entities. To support the data integrity, it combines encrypted unique tags (*UTs*) with the encrypted datasets of all participants. In addition, to retain the anonymity of participants' encrypted data from anyone including authorities, it exploits the re-encryption. Although the proposed framework also practices homomorphic encryption, which usually sacrifices the data integrity, here due to the usage of *UTs*, it is maintained. This paper is a preliminary report of the proposed framework.**

Keywords—Skyline query, ElGamal cryptosystem, Mix-net, Data integrity, Homomorphic encryption

I. INTRODUCTION

In the contemporary world, different commercial organizations, namely hotels, hospitals, real estates, resorts, etc., produce a huge amount of data and use them to make important decisions about their business [1]. Usually, these financial and confidential data are highly sensitive and the organizations do not intend to disclose them in front of other organizations. Hence, maintaining the privacy of these commercial organizations' datasets is essential.

resort	costing	distance
a	20	5
b	10	8
c	20	7
d	30	4
e	10	7
f	30	7

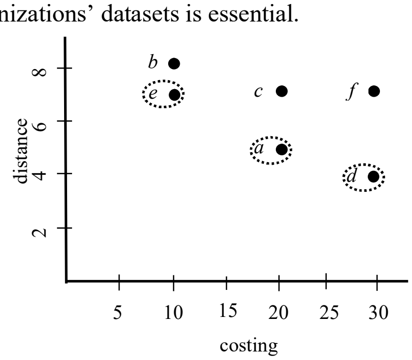

Fig. 1. An example of skyline query for multiple resorts.

Multiple organizations need to analyze their combined dataset to find out those particular ones capable to serve better in their concerned domains. For this, they calculate the selective representative datasets of each participant without disclosing the individual participant's dataset to others. Here, the representative datasets those are not dominated (*i.e.* loser) by other datasets from a large database, is regarded as the skyline [2, 5]. To explain the skyline query, the distances of six resorts, namely *a*, *b*, *c*, *d*, *e*, and *f* are considered from a particular location. Now, based on their costing, a dataset of 2-dimensions (*D*s) is plotted in Fig. 1. For example, between any two arbitrary resorts *i* and *j*, if $i.D_k > j.D_w$ ($k \cong$ {(1 and 2) or 1 or 2}, $w \cong$ {(1 and 2) or 1 or 2}), it implies that *i* is dominated and *j* is non-dominated [9]. The figure shows that from the particular location, the distance of resort *a* is nearer than resort *c* with the same costing. Hence, *c* is dominated by *a*.

This paper develops a framework for multi-party skyline query that confirms both privacy and data integrity. To attain these, it attaches encrypted unique tags (*UT*s) along with the participant's encrypted datasets that confirms the data integrity. Besides, it deploys anonymous credentials [14] to ensure the participant's anonymous identity, commutative ElGamal encryption [4] to assure significant privacy through multiple entities' mutual effort, and re-encryption mix-net [7] to keep the anonymity of participant's encrypted data even from involved authorities, altogether.

The following sections provide a description of the proposed framework. Namely, Section 2 reviews related works. Section 3 illustrates the required cryptographic tools. Section 4 explains the configuration and the individual stages of the framework. Section 5 presents the evaluation and finally, Section 6 concludes the paper.

II. RELATED WORKS

While developing an information security protocol, data integrity is definitely a more basic requirement than data privacy. Due to any reason, if the data integrity is compromised, the originality of the data would be breached. Hence, without retaining data integrity, any attempt to maintain data privacy becomes impractical. Homomorphic encryption (HE) – a widely used cryptographic technique to develop various information security protocols can easily breach data integrity if proper precautions are not taken. For example, in a secure multi-party skyline framework, a dishonest participant may intentionally attempt to avail an illegal benefit by breaching the data integrity. Many existing framework designs did not concentrate on this critical issue.

Namely, the framework proposed in [10] was designed for multiple parties, aimed to ensure privacy, and employed additive HE property of the Paillier cryptosystem. However, it had compromised the data integrity due to the trivial use of HE. Besides, for encryption-decryption operations, the secret decryption key was possessed only by a single entity and did not consider any form of mix-net. Thereby, possibly the dataset can be modified by unfair entities. These deteriorate the practicality as well as efficiency of the framework.

Another skyline framework proposed in [9] executed computation for two parties and was possible to extend into a multi-party platform. Here, the dominance relationship was computed by comparing two individual parties' objects. It developed a protocol named 'Efficient Secure Vector Comparison (ESVC)'. This protocol did not disclose the object's attributes to one another but revealed the dominance relationship between two specific objects. The framework proposed in [13] made a ranking of the attribute value on each dimension of data and used this rank for computation. Still, there exists suspicion that whether the attributes' rank of the objects can ensure privacy or not.

Frameworks proposed in [3, 6, 8] used the data provider which could not know the user's dynamic skyline query. Also, the user could not know the entire private database of the data provider other than the skyline result. An important criterion was most of them included a semi-honest third party to conduct the privacy-preserving skyline query. But it is challenging to assume an unbiased third party as it may involve in the conspiracy. The framework proposed in [1] employed a secure multi-party sorting protocol and semi-honest adversary model. Here, it preserved the order of each attribute for transforming the attribute value of the objects.

Different from existing works, the proposed framework encompasses privacy and data integrity altogether. It attaches encrypted *UT*s along with each participant's encrypted data. Thereby for illegal benefit, if any dishonest entity attempts to breach the data integrity, the dishonesty is finally revealed from disclosed skyline queries. Also, to maintain sufficient privacy it considers no trustworthy entities, *i.e.* trust (secret keys) is distributed among multiple authorities by adopting mix-net consists of at least two mix-nodes.

III. CRYPTOGRAPHIC TOOLS

Major cryptographic tools required to develop the framework are as below. In the following, it is assumed that there exists a mix-net consists of P (≥ 2) mix-nodes and the system manager *SB* is a representative of the mix-net.

A. ElGamal Cryptosystem

ElGamal cryptosystem [15] comprises key generation, encryption, and decryption operations. Here, the sender encrypts its message m and sends it to the receiver.

1) Key generation: The receiver generates a large prime number p and a generator g of the multiplicative group $Z_p{}^*$ of the integers modulo p. Now, it picks a random integer X ($1 < X < p-2$) and computes $Y = g^X \mathit{mod}\, p$. Then, it keeps (X) as the private key and publishes (p, g, Y) as the public key.

2) Encryption: To encrypt m ($0 < m < p-1$), the sender selects a secret random integer k ($1 < k < p-2$) and uses the public key (p, g, Y) to compute $y_1 = g^k \mathit{mod}\, p$ and $y_2 = m.Y^k \mathit{mod}\, p$. Now, it sends $E_Y(k, m) = (y_1, y_2)$ as its encrypted message to the receiver.

3) Decryption: To retrieve the message m from $E_Y(k, m)$, the receiver uses its private decryption key X and computes $m = y_2 \times (y_1{}^X)^{-1} \mathit{mod}\, p$.

B. Commutative encryption, re-encryption and verification

The mix-net can be implemented through commutative encryption technique [4] where it is assumed that the mix-net comprises of P (≥ 2) mix-nodes $M_1, \ldots, M_P$. In the previous section as p and g are defined, X_i and $Y_i = g^{Xi}$ ($\mathit{mod}\ p$) are secret and public keys of each mix-node M_i ($1 < i \leq P$). Now by involving multiple independent mix-nodes, the combined encryption key Y_* is calculated as $Y_* = Y_1 Y_2 \ldots Y_P$ ($\mathit{mod}\ p$) $= g^{X1+\ldots+XP}$ ($\mathit{mod}\ p$). By using Y_*, the encryption of message m is $E_{Y*}(k, m)$, *i.e.* it is identical to the encryption mechanism of section III (A). In the following, notation ($\mathit{mod}\ p$) is omitted.

Later on, $M_1, \ldots, M_P$ conduct the re-encryption operation over $E_{Y*}(k, m)$ as follows.

- The first mix-node M_1 obtains $E_{Y*}(k, m) = \{g^k, m.Y_*{}^k\}$ as the input.
- By using its secret integer k_h ($h \in \{1, \ldots, i$), consecutively i-th M_i calculates $E_{Y*}(k + (\sum_{h=1}^{i} k_h),$ $m) = \{(g^k.(\sum_{h=1}^{i} g^{k_h})), (m.Y_*{}^k.(\sum_{h=1}^{i} Y_*^{k_h}))\}$ and every M_i shuffles the results.
- Thus, lastly M_P produces $E_{Y*}(k_*, m) = \{(g^k.g^{(k1+\ldots+kP)}),$ $(m.Y_*{}^k.Y_*{}^{(k1+\ldots+kP)})\}$ where $k_* = k+k_1+\ldots+k_P$.
- Finally, while decryption, $M_1, \ldots, M_P$ decrypt $E_{Y*}(k_*,$ m) repeatedly by using $X_1, \ldots, X_P$ which is analogous to the decryption mechanism of section III (A).

The following procedure shows– the joining of encrypted *UT*s (described in section III (C)) with participants' datasets, confirms data integrity as well as makes mix-net verifiable without verifying individual mix-nodes. Here, *SB* calculates data-pairs by Cartesian product operation of each two participant's datasets {it will be discussed in (step *e*) of section IV (C) (3)}. Before this operation, each PA_n joins *UT*s with its datasets individually.

- Each PA_n encrypts it's every dataset, then joins a *UT*, *i.e.* $E_{Y*}(r_{*n}, U_n)$ with it through HE.
- Now, through the re-encryption process told above, $M_1, \ldots, M_P$ re-encrypt the encrypted data-pairs and *UT*s along with shuffling, to make them anonymized.
- Later on, $M_1, \ldots, M_P$ repeatedly decrypt each re-encrypted data-pair while required.

Here, no one can link between encrypted and re-encrypted pair-wise datasets due to shuffling. Again, by dividing the data-pair of each PA_n by the *UT* assigned to it previously, the actual dataset is regained. Thus, the honesty of involved entities is confirmed.

C. Unique tags (UTs)

*UT*s are unique and registered integers $U_1, \ldots, U_n$. The notion of *UT*s is identical to confirmation numbers (*CN*s) as in [11, 12]. Here, $r_{n(i)}$ is a secret integer of mix-node M_i, and mix-nodes mutually encrypt each unique tag U_n to $E_{Y*}(r_{*n},$ $U_n) = g^{r*n}, U_n.Y_*{}^{r*n}$ where $r_{*n} = r_{n(1)}+ \ldots + r_{n(P)}$ and shuffle the results. Thereby, it is not possible to link between $E_{Y*}(r_{*n},$ U_n) to U_n. The encryption procedure of *UT*s proceeds as follows.

1. At first SB generates UTs and discloses them publicly, then mix-nodes encrypt them.
2. For encryption, each M_i receives the UTs as $E_{Y*}(r_{*n(i\text{-}1)}, U_n)$ from $M_{i\text{-}1}$. Now, M_i encrypts it to $E_{Y*}(r_{*n(i)}, U_n)$, = $(g^{r*n(i\text{-}1)}g^{rn(i)}, U_nY_*^{r*n(i\text{-}1)}Y_*^{rn(i)})$ and sends the shuffled results to M_{i+1}.
3. While re-encryption, each M_i receives $E_{Y*}(r_{*n}+x_{*n(i\text{-}1)}, U_n)$ from $M_{i\text{-}1}$ and further encrypts it to $E_{Y*}(r_{*n}+x_{*n(i)}, U_n)$ by using the secret integer $x_{n(i)}$, and shuffles the encryption results to forward to M_{i+1}. Then finally, M_P outputs $E_{Y*}(r_{*n}+x_{*n}, U_n)$ where, $x_{*n} = x_{n(1)}+ \ldots + x_{n(P)}$.

Alongside, to enable each registered participant PA_n to show its identity to the SB anonymously, the framework exploits the anonymous credential proposed in [14].

IV. Proposed Framework for Multi-party Skyline Query

This section describes the involved entities and the desired privacy for this work. Fig. 2 depicts the major interactions among entities.

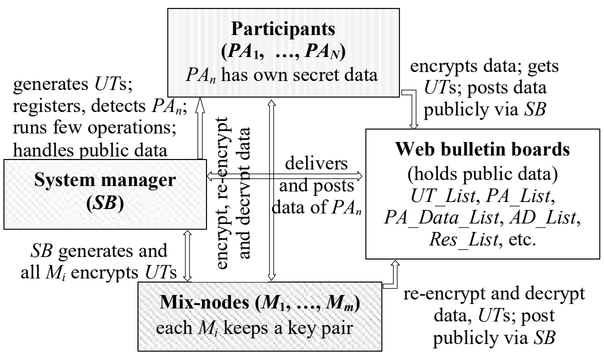

Fig. 2. Major interactions among involved entities.

A. Involved Entities

The entities involved in the developed framework are N (≥ 2) participants (PA_1, …, PA_N), a system manager SB, a mix-net consists of P (≥ 2) mix-nodes, and several public web bulletin boards (WBB) [12]. Their roles are as follows.

***Participant PA_n*:** Each participant PA_n holds confidential datasets of its own organization. Based on datasets of all participants, skyline query result is computed.

System manager SB: SB is considered as a representative of mix-nodes. It performs the registration of participants and later on identifies them anonymously. Besides, it generates UTs, interacts with participants and mix-nodes, performs some non-cryptographic operations, *e.g.* Cartesian products, and posts PA_n's data publicly on WBBs, etc.

***Mix-node M_i*:** Together with other mix-nodes involved in the same mix-net, the main task of any mix-node M_i is to encrypt, re-encrypt (along with shuffling while required), and decrypt the UTs and confidential data of each PA_n. Every M_i keeps its required secret integers, a private and public key-pair (X_i, Y_i) of commutative ElGamal encryption technique.

***Web bulletin board WBB*:** WBB publicly posts important or obligatory data of interactions required for the participants. Namely, UT_List, PA_List, PA_Data_List, AD_List, Res_List, etc., WBBs publicly post the list of UTs in both plain and encrypted forms, the list of participants who have obtained encrypted UTs, encrypted data and UTs of participants, re-encrypted (*i.e.* anonymized) data, and the list of participants' winner data, etc., respectively. Fig. 3 shows the configuration of these WBBs.

PID	signature (S)
…	
IPA_n	PA_n's signature
…	

a) PA_List

UT	encrypted UT	S
…	…	…
U_n	$E_{Y*}(r_{*nj}, U_{nj})$	S_n
…	…	…

b) UT_List

PA_n's data	S
…	
$<E_{Y*}(k_{*nj}+r_{*nj}, DA_{nj}U_{nj}), E_{Y*}(r_{*nj}, U_{nj})>$	S'_n
…	

c) PA_Data_List

re-encrypted data-pair
…
$<\{E_{Y*}(K_{*nj}, DA_{nj}U_{nj}), E_{Y*}(R_{*nj}, U_{nj})\}, \{E_{Y*}(K_{*oq}, DA_{oq}U_{oq}), E_{Y*}(R_{*oq}, U_{oq})\}>$
…

d) AD_List

data-pair	skyline
…	result is in < Table 1 ~ Table 3 >
$<\{DA_{nj}U_{nj}, U_{nj}\}, \{DA_{oq}U_{oq}, U_{oq}\}>$	
…	

e) Res_List

Fig. 3. Configurations of different web bulletin boards.

B. Desired Privacy

For the proposed framework, considerations regarding the desired privacy are as follows.

- Each PA_n keeps the data secret to everyone until encrypting the dataset.
- No participant has any idea about how many data-pairs are owned by other participants.
- When the number of participants are more than two, no participant can know which and how many objects of other participants dominates its' dominated objects.
- No participant can know how many of its objects dominate other participant's objects or not.
- From the final skyline result, as the data is already exposed, anyone can know the winner dataset.

C. Individual Stages

The proposed framework consists of five distinct stages. These are: (1) UT generation, (2) Registration, (3) Data submission, (4) Data anonymization and result revelation, and (5) mismatch detection. They are explained below.

1) UT generation

Major operations of this stage are: (*a*) generation of UTs, (*b*) repeated encryption of UTs, and (*c*) publicly post UTs in both plain and encrypted forms on UT_List. Here, SB and M_1, …, M_P act and interact as follows.

a) First SB generates a total of U integers as $U = \sum_{n=1}^{N} U_n$, $U_n = \sum_{j=1}^{L} U_{nj}$ {($n \in (1, \ldots, N)$), ($j \in (1, \ldots, L)$)}. Thereby, U_{nj} denotes j-th UT of n-th participant (PA_n). Again, $DA_n = \sum_{j=1}^{L} DA_{nj}$, *i.e.* DA_n is the total number of data owned by PA_n, L is the highest range of PA_n's data, and the value of L can be distinct for different PA_n. But the dimension D of datasets of each PA_n must be equal. Hence, each

PA_n needs to declare the number of its total data to SB in advance. Now, SB handovers U to mix-nodes.

b) Based on the mechanism of section III (C), $M_1, \ldots, M_P$ encrypt UTs, *i.e.* converts each U_{nj} to $E_{Y^*}(r_{*nj}, U_{nj})$ through their combined effort. Thereby, no one can identify the link between U_{nj} and $E_{Y^*}(r_{*nj}, U_{nj})$.

c) Finally, SB posts all UTs, *i.e.* U_{nj} as well as $E_{Y^*}(r_{*nj}, U_{nj})$ on *UT_List* publicly.

2) Registration

Major operations of this stage are: (*a*) registration of each PA_n under the supervision of SB, (*b*) delivery of required UTs to PA_n, and (*c*) approval of all assigned UTs by PA_n itself. Here, interactions between SB and PA_n proceed as follows.

a) Each legitimate participant PA_n shows its identity IPA_n to SB through in-person communication.

b) By putting a signature on the designated portion of the *PA_List*, PA_n completes its registration.

c) Next, PA_n obtains an anonymous credential (along with required attributes) from SB. Thereby, later on, PA_n can appear anonymously to SB.

d) Now, SB delivers the required number of encrypted UTs to the registered PA_n which is already specified by itself {discussed in (step *a*) of section IV (C) (1)}.

e) Finally, each PA_n approves its obtained UTs publicly by putting a signature on the designated portion of the *UT_List* as shown in Fig. 3.

3) Data submission

Major operations of this stage are: (*a*) encryption of PA_n's confidential data and attaching with UTs, (*b*) approval of data by PA_n itself, and (*c*) calculation of Cartesian products by SB. Here, SB and PA_n interact as follows.

a) At first each PA_n encrypts its own data ($DA_{n1}, \ldots, DA_{nL}$), *i.e.* calculates $\{E_{Y^*}(k_{*n1}, DA_{n1}), \ldots, E_{Y^*}(k_{*nL}, DA_{nL})\}$ based on the mechanism of section III (B).

b) By using encrypted ($U_{n1}, \ldots, U_{nL}$), *i.e.* $\{E_{Y^*}(r_{*n1}, U_{n1}), \ldots, E_{Y^*}(r_{*nL}, U_{nL})\}$, PA_n calculates $\{E_{Y^*}(k_{*n1}+r_{*n1}, DA_{n1}U_{n1}), \ldots, E_{Y^*}(k_{*nL}+r_{*nL}, DA_{nL}U_{nL})\}$ through the HE operation.

c) Now, PA_n submits $\{[E_{Y^*}(k_{*n1}+r_{*n1}, DA_{n1}U_{n1}), E_{Y^*}(r_{*n1}, U_{n1})], \ldots, [E_{Y^*}(k_{*nL}+r_{*nL}, DA_{nL}U_{nL}), E_{Y^*}(r_{*nL}, U_{nL})]\}$ to SB to be posted publicly on *PA_Data_List*.

d) While PA_n finds its data correctly posted on the *WBB*, for approval it puts another signature on the designated portion of *PA_Data_List* as shown in Fig. 3.

e) Finally, in order to compute the data-pairs of participants' ($PA_1, \ldots, PA_N$) data, SB calculates the Cartesian products of every two participant's datasets. For example, if there are 03 participants PA_1, PA_2 and PA_3 who have A, B and C number of data, respectively; then the total products would be equal to $\{(A \times B) + (A \times C) + (B \times C)\}$. Again, considering DA_{nj} $\{(n \in 1, \ldots, N), (j \in 1, \ldots, L)\}$ as a data of PA_n's dataset, and DA_{oq} $\{(o \in 1, \ldots, N \text{ and } (o \neq n)), (q \in 1, \ldots, L)\}$ as a data of PA_o's dataset; then the computed data-pair is: $\{[E_{Y^*}(k_{*nj}+r_{*nj}, DA_{nj}U_{nj}), E_{Y^*}(r_{*nj}, U_{nj})], [E_{Y^*}(k_{*oq}+r_{*oq}, DA_{oq}U_{oq}), E_{Y^*}(r_{*oq}, U_{oq})]\}$. Thus, the remaining data-pairs are also computed.

4) Data anonymization and result revelation

Major operations of this stage are: (*a*) re-encryption of data along with shuffling, and (*b*) repeated decryption of data by $M_1, \ldots, M_P$. Then, (*c*) initial and final comparison of data-pairs by SB to calculate the skyline query result. Here, SB and $M_1, \ldots, M_P$ continue their works as follows.

a) $M_1, \ldots, M_P$ sequentially re-encrypt and shuffle data-pairs obtained from SB. For example, for the data-pair $\{[E_{Y^*}(k_{*nj}+r_{*nj}, DA_{nj}U_{nj}), E_{Y^*}(r_{*nj}, U_{nj})], [E_{Y^*}(k_{*oq}+r_{*oq}, DA_{oq}U_{oq}), E_{Y^*}(r_{*oq}, U_{oq})]\}$, after applying the re-encryption mechanism through using the secret integers *si* and *ti* of each M_i respectively (as in section III (B)), it is converted into $\{[E_{Y^*}(K_{*nj}, DA_{nj}U_{nj}), E_{Y^*}(R_{*nj}, U_{nj})], [E_{Y^*}(K_{*oq}, DA_{oq}U_{oq}), E_{Y^*}(R_{*oq}, U_{oq})]\}$ = $\{[[g^{(k*nj+r*nj)}.(\sum_{i=1}^{P} g^{si}), (DA_{nj}U_{nj}).Y_*^{(k*nj+r*nj)}.(\sum_{i=1}^{P} Y_*^{si})], [g^{r*nj}.(\sum_{i=1}^{P} g^{si}), (U_{nj}).Y_*^{r*nj}.(\sum_{i=1}^{P} Y_*^{si})]], [[g^{(k*oq+r*oq)}.(\sum_{i=1}^{P} g^{ti}), (DA_{oq}U_{oq}).Y_*^{(k*oq+r*oq)}.(\sum_{i=1}^{P} Y_*^{ti})], [g^{r*oq}.(\sum_{i=1}^{P} g^{ti}), (U_{oq}).Y_*^{r*oq}.(\sum_{i=1}^{P} Y_*^{ti})]]\}$. Where, $\{K_{*nj} = k_{*nj} + r_{*nj} + \sum_{i=1}^{P} si, R_{*nj} = r_{*nj} + \sum_{i=1}^{P} si\}$ and $\{K_{*oq} = k_{*oq} + r_{*oq} + \sum_{i=1}^{P} ti, R_{*oq} = r_{*oq} + \sum_{i=1}^{P} ti\}$. Then, the remaining data-pairs are also re-encrypted in the same way. Alongside, due to the shuffling, the order of incoming and outgoing data-pairs with regard to each M_i becomes irrespective, *i.e.* no link exists in between them which settles 'data anonymization' properly. Lastly, SB posts all re-encrypted data publicly on *AD_List* as in Fig. 3.

b) According to the decryption mechanism discussed in section III (B), $M_1, \ldots, M_P$ use $X_1, \ldots, X_P$ to decrypt all data-pairs sequentially. For example, the final decrypted value of the data-pair $\{[E_{Y^*}(K_{*nj}, DA_{nj}U_{nj}), E_{Y^*}(R_{*nj}, U_{nj})], [E_{Y^*}(K_{*oq}, DA_{oq}U_{oq}), E_{Y^*}(R_{*oq}, U_{oq})]\}$ becomes $\{[(DA_{nj}U_{nj}), U_{nj}], [(DA_{oq}U_{oq}), U_{oq}]\}$. Then, SB publicly posts all decrypted data-pairs on *Res_List* as in Fig. 3.

c) SB retrieves the data-pair as $\{[(DA_{nj}U_{nj}) \,/\, U_{nj}], [(DA_{oq}U_{oq}) \,/\, U_{oq}]\} = \{DA_{nj}, DA_{oq}\}$ (*i.e.* by division) and thus obtains other data-pairs.

d) Over each retrieved data-pair, SB compares ($DA_{nj} > DA_{oq}$), ($DA_{nj} < DA_{oq}$) or ($DA_{nj} \cong DA_{oq}$) to calculate the dominance relationship. Then for other data-pairs, SB continues the same operation. Here, it is assumed that the smaller value of each dimension is better.

e) The initial skyline query result between DA_{ni} and DA_{oq} is denoted as follows.

- DA_{nj} is said to be dominated and DA_{oq} is said to be dominant, *i.e.* $DA_{nj} > DA_{oq}$ if $d_u.DA_{nj} \geq d_u.DA_{oq}$ and $d_u.DA_{ni} > d_u.DA_{oq}$ for at least one dimension d_u ($u \in \{1, \ldots, D\}$) or vice versa.
- DA_{nj} and DA_{oq} are said to be non-dominated to each other *i.e.* $DA_{nj} \cong DA_{oq}$ and both DA_{nj} and DA_{oq} are winners while there exist at least 2 pairs, namely $d_u.DA_{nj} > d_u.DA_{oq}$ and $d_v.DA_{nj} < d_v.DA_{oq}$ ($(u, v) \in \{1, \ldots, D\}$ and $(u \neq v)$).

TABLE I. DATASET FOR TWO PARTICIPANTS

	PA_1	PA_2
DA_1	(1, 5)	(2, 3)
DA_2	(2, 4)	(4, 5)
DA_3	(3, 2)	(5, 1)

- *SB* sets 0 for each of the dominant and 'non-dominated to each other' data (*i.e.* winner), and 1 for the dominated data (*i.e.* loser). Thus, *SB* specifies the dominance relationship between each data-pair of every two participants. Where, for each PA_n, *SB* keeps a track of the initial winner dataset Z_n ($n \in \{1, \ldots, N\}$). At last, *SB* denotes the total number of winner dataset Z as $Z = Z_1 + \ldots + Z_N$. Here, the size of each Z_n is equal to the number of Cartesian products.

TABLE II. INITIAL DOMINANCE RELATIONSHIP FOR TABLE I

Cartesian products	Initially winner	Z_1 for PA_1	Z_2 for PA_2
(1, 5), (2, 3)	(1, 5), (2, 3)	0	0
(1, 5), (4, 5)	(1, 5)	0	1
(1, 5), (5, 1)	(1, 5), (5, 1)	0	0
(2, 4), (2, 3)	(2, 3)	1	0
(2, 4), (4, 5)	(2, 4)	0	1
(2, 4), (5, 1)	(2, 4), (5, 1)	0	0
(3, 2), (2, 3)	(3, 2), (2, 3)	0	0
(3, 2), (4, 5)	(3, 2)	0	1
(3, 2), (5, 1)	(3, 2), (5, 1)	0	0

In order to explain the scenario clearly, the datasets of the 2 participants shown in Table I are considered. The initial skyline query result is shown in Table II. Here, if the redundancy of the winner data is removed, then the winners of the initial comparisons are $Z_1 = (PA_1DA_1, PA_1DA_2, PA_1DA_3)$, and $Z_2 = (PA_2DA_1, PA_2DA_3)$. Thus the values are $Z_1 = 3$, $Z_2 = 2$, then $Z = Z_1 + Z_2 = 5$.

f) The final skyline query result from the initial winner datasets of winner participants A ($A \leq N$) is evaluated as follows. While finally comparing the initial winner datasets, if each winner data $PA_nDA_B = 0$ (($n \in \{1, \ldots, A\}$), ($B \in \{1, \ldots, Z_n\}$), ($B \leq L$)) for other winner datasets $\sum_{o=1,q=1}^{N,Z_o} PA_oDA_q$ ($n \neq o$), then PA_nDA_B is said to be dominant or winner data finally, the combined dominance relationship results for PA_nDA_B against other winner dataset is 0. A single part of the final result for Table I is shown in Table III. Here, PA_1DA_1 is dominant to the initial winner PA_2DA_1 and PA_2DA_3. Thus, PA_1DA_1 gains 0 against the combined dominance relationship of both of them. Hence, PA_1DA_1 is one of the final winner datasets. But PA_1DA_2 is not a final one because it gains 1 against the combined dominance relationship of PA_2DA_1 and PA_2DA_3. Thus, all the final winner datasets among all the initial winner participants are calculated.

TABLE III. A SINGLE PART OF FINAL DOMINANCE RELATIONSHIP BETWEEN INITIAL WINNER PARTICIPANTS

($PA_2 > PA_1$)?	PA_1DA_1	PA_1DA_2	PA_1DA_3
$PA_2DA_1 >$	0	1	0
$PA_2DA_3 >$	0	0	0
$\sum(PA_2DA_1 > + PA_2DA_3 >)$	0	1	0

5) Mismatch detection

This stage is merged with step *b* of the data anonymization and result revelation stage. Any data on the decrypted data-pair of the *Res_List* is regarded as inconsistent through the following identification.

- At first, $M_1, \ldots, M_P$ construct a list of used *UT*s $L(\cup\Gamma)$ by decrypting the encrypted *UT*s approved by every registered PA_n on the *UT_List*.
- While a comparison between decrypted *UT*s (those are with data-pairs) of *Res_List* and $L(\cup\Gamma)$ is found as the same, it implies that there is no inconsistency.
- Otherwise, the mismatch can be identified, namely when: (*i*) the same *UT* appears twice on *Res_List*, or (*ii*) a *UT* appears on *Res__List* but does not exist $L(\cup\Gamma)$, or (*iii*) the number of *UT*s (along with data-pairs) on *Res_List* is more or less than the number of *UT*s in $L(\cup\Gamma)$, etc. These imply that $M_1, \ldots, M_P$ are dishonest.

If dishonesty is identified once, $M_1, \ldots, M_P$ are asked to repeat the reverse operation to reach the data of the *PA_Data_List*, analogous to the procedure discussed in [12]. While a mismatch is identified, the liable mix-node is detected. Actually, in a real sense, the mismatch detection stage is not for detecting dishonesty, instead, it ensures the honesty of involved entities.

V. PRELIMINARY EVALUATION

The proposed framework maintains the following requirements.

Privacy: Each PA_n encrypts its own data and $M_1, \ldots, M_P$ encrypt *UT*s using the combined encryption key Y_* which confirms the privacy of data sufficiently. Also, while PA_n submits its data, the *SB* authenticates PA_n anonymously through PA_n's anonymous credential. Thus, the privacy of both PA_n's data and the identity of PA_n itself is maintained. In addition, no one can know the link between PA_n and its encrypted data. Thus, the framework maintains privacy systematically in multiple ways.

Data integrity: As already discussed, while a framework is developed, due to the usage of HE operation, in many cases, the data integrity is breached or sacrificed. But herein, in order to maintain the data integrity sincerely, it attaches registered *UT*s along with PA_n's data individually. Besides, the data of interactions of all PA_n are exposed publicly on various *WBB*s. Thereby, if the data integrity is compromised, anyone can detect it from the publicly disclosed data. Also, as discussed in section IV (C) (5), for any such case, the entity liable for it is also identifiable.

Accuracy: Only the legitimate participant can involve in the framework as they need to be registered first. Then, only the registered PA_n obtains the required number of *UT*s and attaches them separately with its dataset. All datasets including the PA_n's submitted ones are posted on different public *WBB*s. After calculating the initial and the final skyline results which are also published on *WBB*, anyone can check the accuracy of the result.

Data anonymization: Participants' encrypted data posted on *PA_Data_List* are re-encrypted and shuffled by multiple entities prior to being posted on *AD_List*. Thereby, no one can know any link between data on *PA_Data_List* and decrypted data with skyline results are posted on *Res_List*.

Based on adopted cryptographic schemes and other vital aspects, Table IV presents a comparison between the proposed framework and the one proposed in [10]. Where framework [10] was also designed for a multi-party skyline query and targeted to retain privacy. However, the table implies that various aspect considered by the proposed framework is more practical than those of [10].

TABLE IV. COMPARISON BASED ON CRYPTOGRAPHIC SCHEMES AND OTHER MAJOR ASPECTS

Aspects	**Framework**	
	Proposed	***[10]***
How to authenticate registered participants	by anonymous credential	not mentioned
How encryption key is generated	by combined encryption key of multiple entities	by individual key of the participant
Exploited cryptosystem	ElGamal	Paillier
How many entities hold private decryption key	multiple ($\geq$ 2) mutual independent entities	only the single entity (participant)
Who conducts skyline queries	a third party who doesn't need to be trusted	involved participants
Storage of data	on public *WBB*s	participant itself
How to maintain data integrity	by attaching *UT*s with data	not considered
How to anonymize encrypted data	re-encryption and shuffling of the mix-net	XOR operation and permutation

VI. CONCLUSIONS

The proposed framework for multi-party skyline query exploits a commutative encryption technique by adopting multiple mutual entities. Thereby, the data encryption-decryption procedure becomes sufficiently robust. Most importantly, it attaches *UT*s to participants' individual data that maintains data integrity. In addition, the re-encryption operation over encrypted data makes them anonymized. The preliminary evaluation implies that the framework is more practical than the existing ones. As for upcoming works, procedures in individual stages must be enhanced, and volumes of computations and communications must be evaluated while developing a prototype system.

REFERENCES

[1] M. Qaosar, A. Zaman, M. A. Siddique, A. Annisa, and Y. Morimoto, "Privacy-Preserving Secure Computation of Skyline Query in Distributed Multi-Party Databases," Information (Switzerland), Vol. 10, No. 03, p. 119, 2019.

[2] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in 17th Int. Conf. on data engineering. IEEE, pp. 421–430, 2001.

[3] W. Chen, M. Liu, R. Zhang, Y. Zhang, and S. Liu, "Secure outsourced skyline query processing via untrusted cloud service providers," in 35th Annual IEEE International Conference on Computer Communications. IEEE INFOCOM, pp. 1 – 9, 2016.

[4] N. Islam, K. M. R. Alam, and S. S. Rahman, "Commutative re-encryption techniques: Significance and analysis," Information Security Journal: A Global Perspective, Taylor & Francis, Vol. 24, No. 4-6, pp. 185-193, 2015.

[5] X. Han, J. Li, D. Yang, and J. Wang, "Efficient skyline computation on big data," IEEE Trans. on Knowledge and Data Engineering, Vol. 25, No. 11, pp. 2521–2535, 2012.

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

[7] N. Islam, K. M. R. Alam, and A. Rahman, "The effectiveness of mixnets–an empirical study," Computer Fraud & Security, Elsevier, Vol. 2013, No. 12, pp. 9–14, 2013.

[8] J. Liu, J. Yang, L. Xiong, and J. Pei, "Secure skyline queries on cloud platform," in 2017 IEEE 33rd Int. Conf. on data engineering (ICDE). IEEE, pp. 633–644, 2017.

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

[10] M. Qaosar, K. M R. Alam, A. Zaman, C. Li, S. Ahmed, M. A. Siddique, and Y. Morimoto, "A framework for privacy-preserving multi-party skyline query based on homomorphic encryption," IEEE Access Vol. 7, pp. 167481–167496, 2019.

[11] K. M. R. Alam, S. Tamura, S. Taniguchi, and T. Yanase, "An anonymous voting scheme based on confirmation numbers," IEEJ Trans. on EIS, Vol. 130, No. 11, pp. 2065–2073, 2010.

[12] K. M. R. Alam, S. Tamura, S. M. S. Rahman, and Y. Morimoto, "An electronic voting scheme based on revised-SVRM and confirmation numbers," IEEE Trans. on Dependable and Secure Computing, Vol. 18, No. 1, pp. 400-410, 2021.

[13] A. Zaman, M A. Siddique, Y. Morimoto, "Secure computation of skyline query in mapreduce," in Int. Conf. on Advanced Data Mining and Applications. Springer, pp. 345–360, 2016.

[14] S. Tamura and S. Taniguchi, "Enhancement of anonymous tag based credentials," in Information Security Comput. Fraud, Vol. 2, No. 1, pp. 10–20, 2014.

[15] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Trans. on Information Theory, Vol. 31, No. 4, pp. 469.472, 1985.