Secure Skyline Queries on Encrypted Data

CS 573 Data Privacy and Security

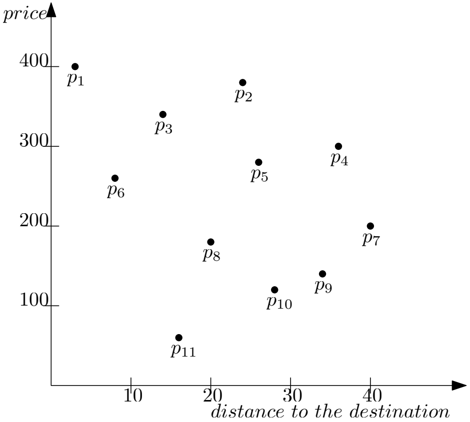
Jinfei Liu, Juncheng Yang, Li Xiong, and Jian Pei. Secure Skyline Queries on Cloud Platform. ICDE 2017.
Jinfei Liu, Juncheng Yang, Li Xiong, and Jian Pei. Secure and Efficient Skyline Queries on Encrypted Data. TKDE 2018.

2018-11-19

EMORY

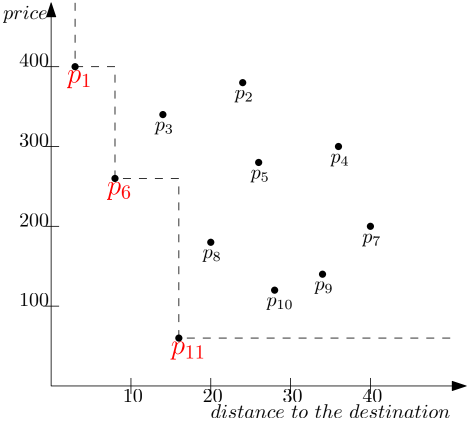
Skyline Computation: Hotel Example

hotel	*distance*	*price*
p_1	4	400
p_2	24	380
p_3	14	340
p_4	36	300
p_5	26	280
p_6	8	260
p_7	40	200
p_8	20	180
p_9	34	140
p_{10}	28	120
p_{11}	16	60



Skyline Computation: Hotel Example

hotel	*distance*	*price*
p_1	4	400
p_2	24	380
p_3	14	340
p_4	36	300
p_5	26	280
p_6	8	260
p_7	40	200
p_8	20	180
p_9	34	140
p_{10}	28	120
p_{11}	16	60



Motivating Example: Skyline Queries

Table: Sample of heart disease dataset.

(a) Original data.

ID	age	trestbps
$\mathbf{p}_1$	40	140
$\mathbf{p}_2$	39	120
$\mathbf{p}_3$	45	130
$\mathbf{p}_4$	37	140

(b) Mapped Data.

ID	age	trestbps
$\mathbf{t}_1$		
$\mathbf{t}_2$		
$\mathbf{t}_3$		
$\mathbf{t}_4$		

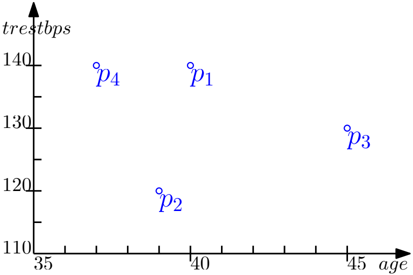


Figure: q(41,125).

Motivating Example: Skyline Queries

Table: Sample of heart disease dataset.

(a) Original data.

ID	age	trestbps
$\mathbf{p}_1$	40	140
$\mathbf{p}_2$	39	120
$\mathbf{p}_3$	45	130
$\mathbf{p}_4$	37	140

(b) Mapped Data.

ID	age	trestbps
$\mathbf{t}_1$	42	140
$\mathbf{t}_2$	43	130
$\mathbf{t}_3$	45	130
$\mathbf{t}_4$	45	140

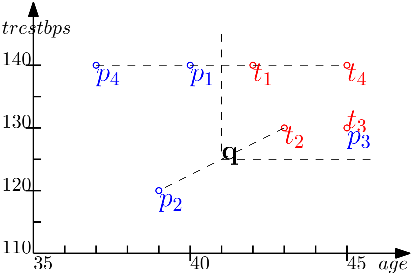


Figure: q(41,125).

Motivating Example: Skyline Queries

Table: Sample of heart disease dataset.

(a) Original data.

ID	age	trestbps
$\mathbf{p}_1$	40	140
$\mathbf{p}_2$	39	120
$\mathbf{p}_3$	45	130
$\mathbf{p}_4$	37	140

(b) Mapped Data.

ID	age	trestbps
$\mathbf{t}_1$	42	140
$\mathbf{t}_2$	43	130
$\mathbf{t}_3$	45	130
$\mathbf{t}_4$	45	140

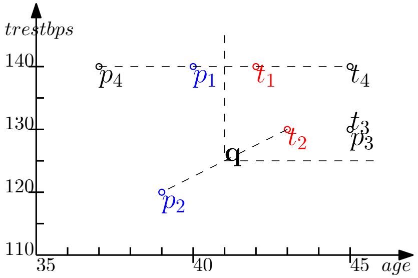
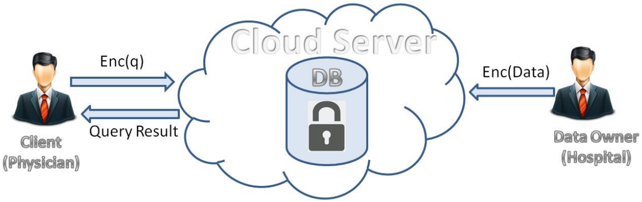


Figure: q(41,125).

Secure Similarity Queries



Related Work

- Fully homomorphic encryption - impractical

Related Work

- Fully homomorphic encryption - impractical
- Order preserving encryption - subjective to attacks

Related Work

- Fully homomorphic encryption - impractical
- Order preserving encryption - subjective to attacks
- Partially homomorphic encryption - limited computation but efficient, many focused on knn queries, challenging for skyline due to complex comparisons

Outline

- Problem setting

Outline

- Problem setting
- Paillier crypto scheme

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol

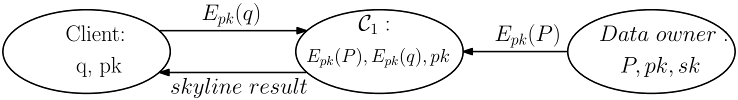
Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

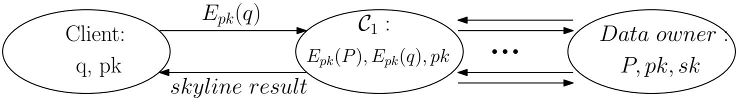
Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

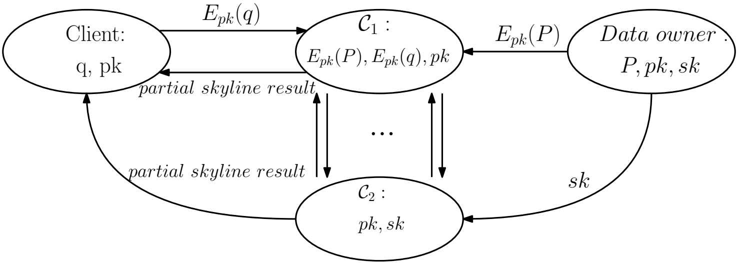
Problem Setting



Problem Setting

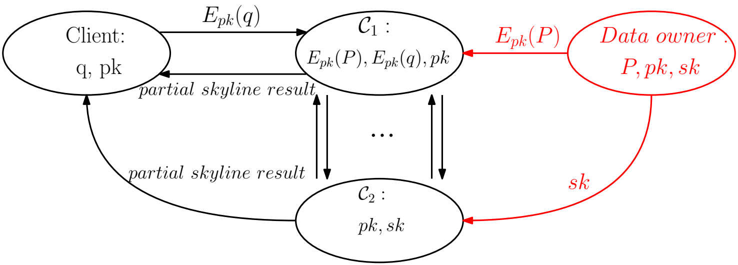


Problem Setting



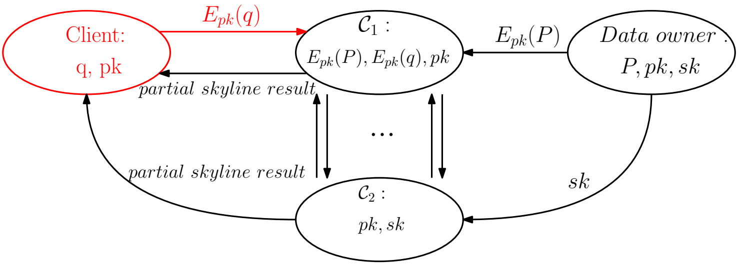
$\mathcal{C}_1$ and $\mathcal{C}_2$ are non-colluding

Problem Setting



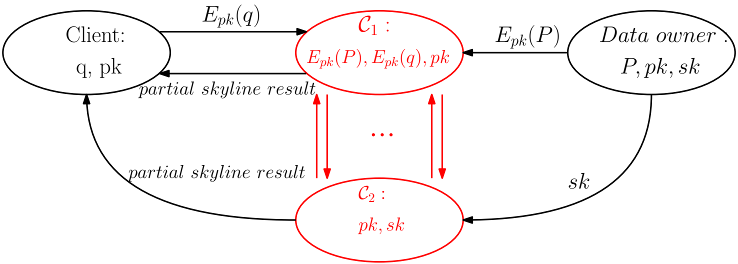
Data owner (e.g., hospital, CDC) sends private key to $\mathcal{C}_2$.
Data owner sends $E_{pk}(\mathbf{p}_i[j])$ for $i = 1, ..., n$ and $j = 1, ..., m$ to cloud server $\mathcal{C}_1$.

Problem Setting



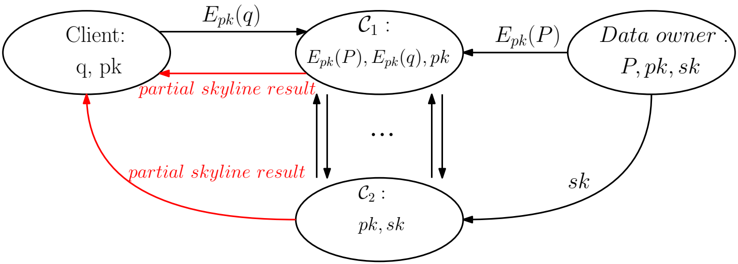
An authorized client (e.g., physician) sends $E_{pk}(\mathbf{q})$ to cloud server $\mathcal{C}_1$.

Problem Setting



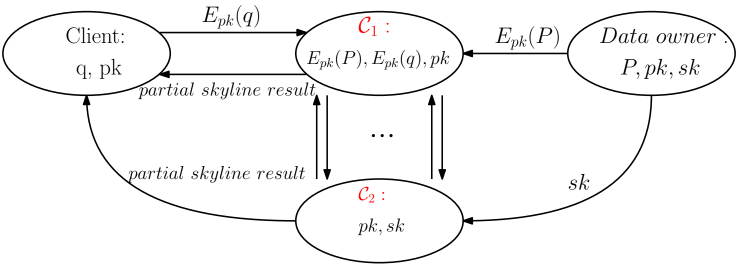
Our goal is to enable the cloud server to **compute** and return the skyline to the client without learning any information about the data and the query.

Problem Setting



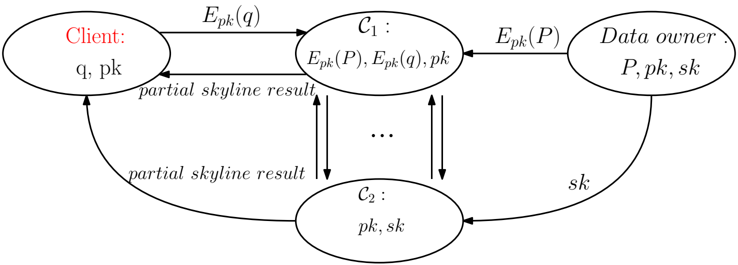
Our goal is to enable the cloud server to compute and **return** the skyline to the client without learning any information about the data and the query.

Problem Setting: Desired Privacy Properties



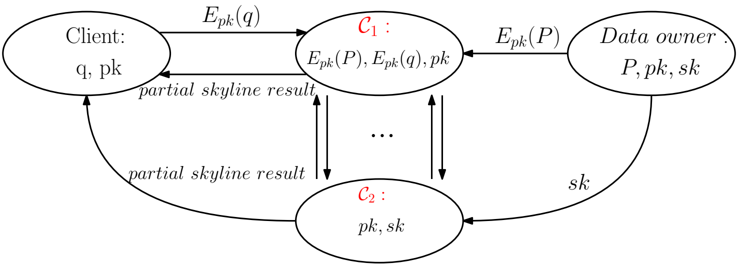
Data Privacy. Cloud servers $\mathcal{C}_1$ and $\mathcal{C}_2$ know nothing about the exact data except the size pattern, the client knows nothing about the dataset except the skyline query result.

Problem Setting: Desired Privacy Properties



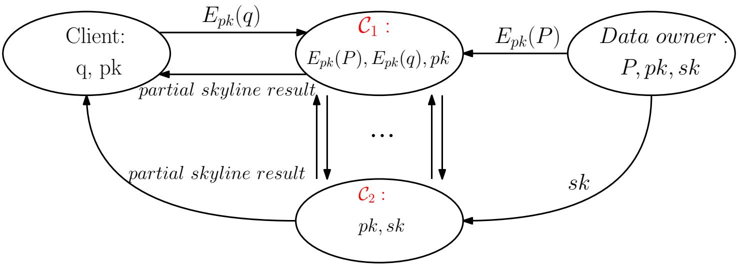
Data Privacy. Cloud servers $\mathcal{C}_1$ and $\mathcal{C}_2$ know nothing about the exact data except the size pattern, the client knows nothing about the dataset except the skyline query result.

Problem Setting: Desired Privacy Properties



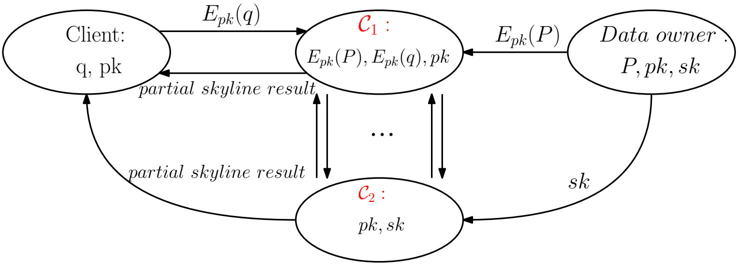
Data Pattern Privacy. Cloud servers $\mathcal{C}_1$ and $\mathcal{C}_2$ know nothing about the data patterns (indirect data knowledge) due to intermediate result, e.g., which tuple dominates which other tuple.

Problem Setting: Desired Privacy Properties



Query Privacy. Data owner, cloud servers $\mathcal{C}_1$ and $\mathcal{C}_2$ know nothing about the query tuple **q**.

Problem Setting: Desired Privacy Properties



Result Privacy. Cloud servers $\mathcal{C}_1$ and $\mathcal{C}_2$ know nothing about the query result, e.g., which tuples are in the skyline result.

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

Paillier Cryptosystem

- Homomorphic addition of plaintexts:

$$D_{sk}(E_{pk}(a) \times E_{pk}(b) \ mod \ N^2) = (a + b) \ mod \ N$$

- Homomorphic multiplication of plaintexts:

$$D_{sk}(E_{pk}(a)^b \ mod \ N^2) = a \times b \ mod \ N$$

https://mhe.github.io/jspaillier/

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

Basic Security Subprotocols: Secure Multiplication (SM)

- Input
  - $\mathcal{C}_1$: encrypted input $E_{pk}(a)$ and $E_{pk}(b)$
  - $\mathcal{C}_2$: private key sk
- Output
  - $\mathcal{C}_1$ knows $E_{pk}(a \times b)$
  - $\mathcal{C}_2$ knows nothing

Basic Security Subprotocols: Secure Bit Decomposition (SBD)

- Input
  - $\mathcal{C}_1$: encrypted input $E_{pk}(a)$
  - $\mathcal{C}_2$: private key sk
- Output
  - $\mathcal{C}_1$ knows encrypted individual bits of the binary representation of a, denoted as $[\![a]\!] = \langle E_{pk}((a)_B^{(1)}), ..., E_{pk}((a)_B^{(l)})\rangle$, where l is the number of bits, $(a)_B^{(1)}$ and $(a)_B^{(l)}$ denote the most and least significant bits of a, respectively.
  - $\mathcal{C}_2$ knows nothing

Basic Security Subprotocols

- Secure OR (SOR)
- Secure AND (SAND)
- Secure NOT (SNOT)
- Secure Less Than or Equal (SLEQ)
- Secure Equal (SEQ)
- Secure Less (SLESS)
- Secure Minimum (SMIN)

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

Challenge of Secure Dominance Protocol

- For each comparison between two tuples $\mathbf{p}_a$ and $\mathbf{p}_b$, we need to compare all their m attributes and for comparison of each attribute $\mathbf{p}[j]$, there are three different outputs, i.e., $\mathbf{p}_a[j] < (=, >)\ \mathbf{p}_b[j]$.

Challenge of Secure Dominance Protocol

- For each comparison between two tuples $\mathbf{p}_a$ and $\mathbf{p}_b$, we need to compare all their m attributes and for comparison of each attribute $\mathbf{p}[j]$, there are three different outputs, i.e., $\mathbf{p}_a[j] < (=, >)\ \mathbf{p}_b[j]$.
- Therefore, there are 3^m different outputs for each comparison between two tuples, based on which we need to determine if one tuple dominates the other.

Secure Dominance Protocol

Algorithm 1 Secure Dominance Protocol.

Input: $\mathcal{C}_1$ has $E_{pk}(\mathbf{a}), E_{pk}(\mathbf{b})$ and $\mathcal{C}_2$ has sk.
Output: $\mathcal{C}_1$ gets $E_{pk}(1)$ if $\mathbf{a} \prec \mathbf{b}$, otherwise, $\mathcal{C}_1$ gets $E_{pk}(0)$.
$\mathcal{C}_1$ and $\mathcal{C}_2$:
for $j = 1$ to m **do**
    $\mathcal{C}_1$ gets $\delta_j = E_{pk}(Bool(\mathbf{a}[j] \leq \mathbf{b}[j]))$ by SLEQ
end for
use SAND to compute $\Phi = \delta_1 \wedge ..., \wedge \delta_m$
$\mathcal{C}_1$:
compute $\alpha = E_{pk}(\mathbf{a}[1]) \times, ..., \times E_{pk}(\mathbf{a}[m])$
compute $\beta = E_{pk}(\mathbf{b}[1]) \times, ..., \times E_{pk}(\mathbf{b}[m])$
$\mathcal{C}_1$ and $\mathcal{C}_2$:
$\mathcal{C}_1$ gets $\sigma = E_{pk}(Bool(\alpha < \beta))$ by employing SLESS
$\mathcal{C}_1$ gets $\Psi = \sigma \wedge \Phi$ as the final dominance relationship using SAND

Example of Secure Dominance Protocol.

Algorithm 2 Secure Dominance Protocol.

Input: $\mathcal{C}_1$ has $E_{pk}(\mathbf{a}), E_{pk}(\mathbf{b})$ and $\mathcal{C}_2$ has sk.**a**=(2,5); **b**=(4,5)
Output: $\mathcal{C}_1$ gets $E_{pk}(1)$ if $\mathbf{a} \prec \mathbf{b}$, otherwise, $\mathcal{C}_1$ gets $E_{pk}(0)$.
$\mathcal{C}_1$ and $\mathcal{C}_2$:
for $j = 1$ to m **do**
    $\mathcal{C}_1$ gets $\delta_j = E_{pk}(Bool(\mathbf{a}[j] \leq \mathbf{b}[j]))$ by SLEQ $\delta_1 = 1; \delta_2 = 1$
end for
use SAND to compute $\Phi = \delta_1 \wedge ..., \wedge \delta_m$ $\Phi = 1$
$\mathcal{C}_1$:
compute $\alpha = E_{pk}(\mathbf{a}[1]) \times, ..., \times E_{pk}(\mathbf{a}[m])$ $\alpha = 7$
compute $\beta = E_{pk}(\mathbf{b}[1]) \times, ..., \times E_{pk}(\mathbf{b}[m])$ $\beta = 9$
$\mathcal{C}_1$ and $\mathcal{C}_2$:
$\mathcal{C}_1$ gets $\sigma = E_{pk}(Bool(\alpha < \beta))$ by employing SLESS $\sigma = 1$
$\mathcal{C}_1$ gets $\Psi = \sigma \wedge \Phi$ as the final dominance relationship using SAND $\Psi = 1$

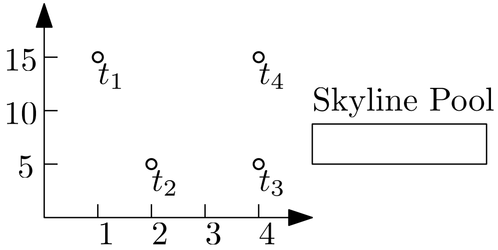
Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

Skyline Computation Algorithm

Algorithm 3 Skyline Computation.

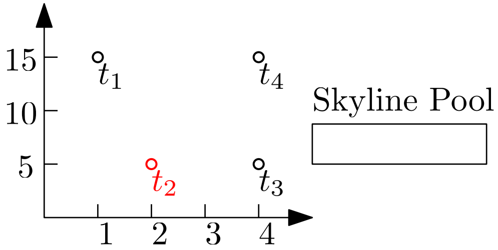
```
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty do
    for i = 1 to size of dataset T do
        S(t_i) = \sum_{j=1}^{m} t_i[j]
        choose the tuple t_min with smallest S(t_i) as a skyline
        add t_min to skyline pool
        delete those tuples dominated by t_min from T
        delete tuple t_min from T
    end for
end while
return skyline pool
```



Skyline Computation Algorithm

Algorithm 4 Skyline Computation.

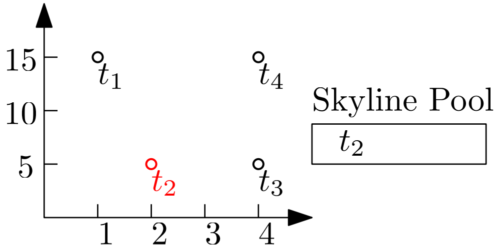
```
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty do
    for i = 1 to size of dataset T do
        S(t_i) = sum_{j=1}^{m} t_i[j]
        choose the tuple t_min with smallest S(t_i) as a skyline
        add t_min to skyline pool
        delete those tuples dominated by t_min from T
        delete tuple t_min from T
    end for
end while
return skyline pool
```



Skyline Computation Algorithm

Algorithm 5 Skyline Computation.

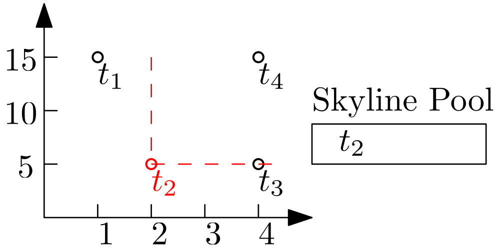
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty **do**
    for $i = 1$ to size of dataset T **do**
        $S(\mathbf{t}_i) = \sum_{j=1}^{m} \mathbf{t}_i[j]$
        choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline
        add $\mathbf{t}_{min}$ to skyline pool
        delete those tuples dominated by $\mathbf{t}_{min}$ from T
        delete tuple $\mathbf{t}_{min}$ from T
    end for
end while
return skyline pool



Skyline Computation Algorithm

Algorithm 6 Skyline Computation.

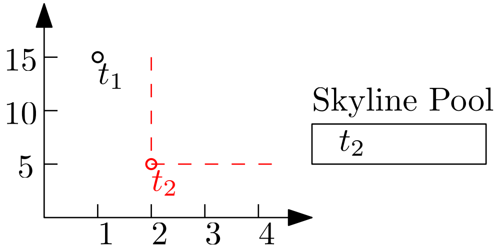
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty **do**
    for $i = 1$ to size of dataset T **do**
        $S(\mathbf{t}_i) = \sum_{j=1}^{m} \mathbf{t}_i[j]$
        choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline
        add $\mathbf{t}_{min}$ to skyline pool
        delete those tuples dominated by $\mathbf{t}_{min}$ from T
        delete tuple $\mathbf{t}_{min}$ from T
    end for
end while
return skyline pool



Skyline Computation Algorithm

Algorithm 7 Skyline Computation.

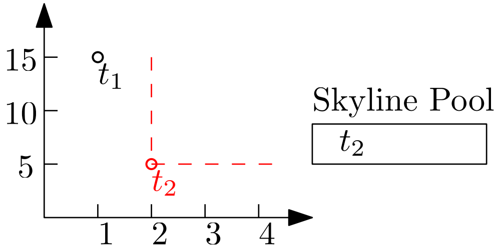
```
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty do
    for i = 1 to size of dataset T do
        S(t_i) = Σ_{j=1}^{m} t_i[j]
        choose the tuple t_min with smallest S(t_i) as a skyline
        add t_min to skyline pool
        delete those tuples dominated by t_min from T
        delete tuple t_min from T
    end for
end while
return skyline pool
```



Skyline Computation Algorithm

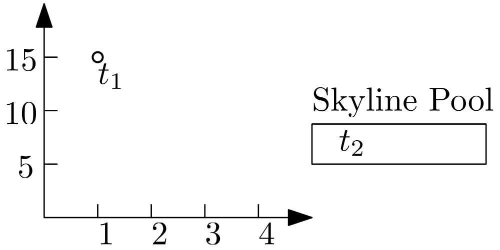
Algorithm 8 Skyline Computation.

```
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty do
    for i = 1 to size of dataset T do
        S(t_i) = Σ_{j=1}^{m} t_i[j]
        choose the tuple t_min with smallest S(t_i) as a skyline
        add t_min to skyline pool
        delete those tuples dominated by t_min from T
        delete tuple t_min from T
    end for
end while
return skyline pool
```



Algorithm 9 Skyline Computation.

```
Input: A dataset T.
Output: Skyline of T.
while the dataset T is not empty do
    for i = 1 to size of dataset T do
        S(t_i) = ∑_{j=1}^{m} t_i[j]
        choose the tuple t_min with smallest S(t_i) as a skyline
        add t_min to skyline pool
        delete those tuples dominated by t_min from T
        delete tuple t_min from T
    end for
end while
return skyline pool
```



choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

Initial case $\mathbf{t}_i$

Party $\mathcal{C}_1$		$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$		
$\mathbf{t}_1$	$(1, 15)$		
$\mathbf{t}_2$	$(2, 5)$		
$\mathbf{t}_3$	$(4, 5)$		
$\mathbf{t}_4$	$(4, 15)$		

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$E_{pk}(S(\mathbf{t}_i)) = E_{pk}(\mathbf{t}_i[1]) \times \ldots \times E_{pk}(\mathbf{t}_i[m]) \mod N^2$$

Party $\mathcal{C}_1$		
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$
$\mathbf{t}_1$	$(1, 15)$	16
$\mathbf{t}_2$	$(2, 5)$	7
$\mathbf{t}_3$	$(4, 5)$	9
$\mathbf{t}_4$	$(4, 15)$	19

$\mathcal{C}_2$	

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$[\![E_{pk}(S(\mathbf{t}_i))]\!] = SBD(E_{pk}(S(\mathbf{t}_i)))$$

Party $\mathcal{C}_1$				$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$		
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$		
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$		
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$		
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$		

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

add a $\lceil logn \rceil$ − bit sequence to the end of each $E_{pk}(S(\mathbf{t}_i))$

$Party\ \mathcal{C}_1$				
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	*pert.*
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$	$1, 1$
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$	$1, 0$
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$	$0, 1$
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$	$0, 0$

$\mathcal{C}_2$	

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

perturbed values guaranteed to be different while order is preserved

Party $\mathcal{C}_1$					
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	*pert.*	$S(\mathbf{t}_i)$
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$	$1, 1$	67
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$	$1, 0$	30
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$	$0, 1$	37
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$	$0, 0$	76

$\mathcal{C}_2$	

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

finding smallest $S(\mathbf{t}_i)$

Party $\mathcal{C}_1$					
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	*pert.*	$S(\mathbf{t}_i)$
$\mathbf{t}_1$	$(1, 15)$	16	1, 0, 0, 0, 0	1, 1	67
$\mathbf{t}_2$	$(2, 5)$	7	0, 0, 1, 1, 1	1, 0	30
$\mathbf{t}_3$	$(4, 5)$	9	0, 1, 0, 0, 1	0, 1	37
$\mathbf{t}_4$	$(4, 15)$	19	1, 0, 0, 1, 1	0, 0	76

$\mathcal{C}_2$	

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$E_{pk}(S(\mathbf{t}_{min}))^{N-1} \times E_{pk}(S(\mathbf{t}_i)) \mod N^2$$

Party $\mathcal{C}_1$						
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	$pert.$	$S(\mathbf{t}_i)$	$S(\mathbf{t}_i) - S(\mathbf{t}_{min})$
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$	$1, 1$	67	$67 - 30$
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$	$1, 0$	30	$30 - 30$
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$	$0, 1$	37	$37 - 30$
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$	$0, 0$	76	$76 - 30$

$\mathcal{C}_2$	

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

randomly noise vector r

$Party\ \mathcal{C}_1$							
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	$pert.$	$S(\mathbf{t}_i)$	$S(\mathbf{t}_i) - S(\mathbf{t}_{min})$	r
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$	$1, 1$	67	$67 - 30$	3
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$	$1, 0$	30	$30 - 30$	9
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$	$0, 1$	37	$37 - 30$	31
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$	$0, 0$	76	$76 - 30$	2

$\mathcal{C}_2$	

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

permutation sequence π

$Party\ \mathcal{C}_1$									$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	$pert.$	$S(\mathbf{t}_i)$	$S(\mathbf{t}_i) - S(\mathbf{t}_{min})$	r	π		
$\mathbf{t}_1$	$(1, 15)$	16	$1, 0, 0, 0, 0$	$1, 1$	67	$67 - 30$	3	2		
$\mathbf{t}_2$	$(2, 5)$	7	$0, 0, 1, 1, 1$	$1, 0$	30	$30 - 30$	9	1		
$\mathbf{t}_3$	$(4, 5)$	9	$0, 1, 0, 0, 1$	$0, 1$	37	$37 - 30$	31	4		
$\mathbf{t}_4$	$(4, 15)$	19	$1, 0, 0, 1, 1$	$0, 0$	76	$76 - 30$	2	3		

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$\pi(E_{pk}(S(\mathbf{t}_{min}))^{N-1} \times E_{pk}(S(\mathbf{t}_i)))^{r_i}$$

Party $\mathcal{C}_1$									$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	*pert.*	$S(\mathbf{t}_i)$	$S(\mathbf{t}_i) - S(\mathbf{t}_{min})$	r	π	β'	
$\mathbf{t}_1$	$(1, 15)$	16	1, 0, 0, 0, 0	1, 1	67	67 − 30	3	2	0	
$\mathbf{t}_2$	$(2, 5)$	7	0, 0, 1, 1, 1	1, 0	30	30 − 30	9	1	111	
$\mathbf{t}_3$	$(4, 5)$	9	0, 1, 0, 0, 1	0, 1	37	37 − 30	31	4	92	
$\mathbf{t}_4$	$(4, 15)$	19	1, 0, 0, 1, 1	0, 0	76	76 − 30	2	3	217	

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$\text{if } \beta'_i = 0, U_i = E_{pk}(1)$$

Party $\mathcal{C}_1$									$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	$S(\mathbf{t}_i)$	$[[S(\mathbf{t}_i)]]$	$pert.$	$S(\mathbf{t}_i)$	$S(\mathbf{t}_i) - S(\mathbf{t}_{min})$	r	π	β'	U
$\mathbf{t}_1$	$(1, 15)$	16	1, 0, 0, 0, 0	1, 1	67	$67 - 30$	3	2	0	1
$\mathbf{t}_2$	$(2, 5)$	7	0, 0, 1, 1, 1	1, 0	30	$30 - 30$	9	1	111	0
$\mathbf{t}_3$	$(4, 5)$	9	0, 1, 0, 0, 1	0, 1	37	$37 - 30$	31	4	92	0
$\mathbf{t}_4$	$(4, 15)$	19	1, 0, 0, 1, 1	0, 0	76	$76 - 30$	2	3	217	0

Secure Skyline Protocol

choose the tuple $\mathbf{t}_{min}$ with smallest $S(\mathbf{t}_i)$ as a skyline

$$V = \pi'(U)$$

Party $\mathcal{C}_1$		
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V
$\mathbf{t}_1$	$(1, 15)$	0
$\mathbf{t}_2$	$(2, 5)$	1
$\mathbf{t}_3$	$(4, 5)$	0
$\mathbf{t}_4$	$(4, 15)$	0

$\mathcal{C}_2$	

Secure Skyline Protocol

add skyline tuple to skyline pool

$$\mathbf{t}_i'[j] = V_i \times \mathbf{t}_i[j]$$

Party $\mathcal{C}_1$				$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V	$(\mathbf{t}_i[1]', \mathbf{t}_i[2]')$		
$\mathbf{t}_1$	$(1, 15)$	0	$(0, 0)$		
$\mathbf{t}_2$	$(2, 5)$	1	$(2, 5)$		
$\mathbf{t}_3$	$(4, 5)$	0	$(0, 0)$		
$\mathbf{t}_4$	$(4, 15)$	0	$(0, 0)$		

add skyline tuple to skyline pool

$$\mathbf{p}_i'[j] = V_i \times \mathbf{p}_i[j]$$

Party $\mathcal{C}_1$				
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V	$(\mathbf{t}_i[1]', \mathbf{t}_i[2]')$	$(\mathbf{p}_i[1]', \mathbf{p}_i[2]')$
$\mathbf{t}_1$	$(1, 15)$	0	$(0, 0)$	$(0, 0)$
$\mathbf{t}_2$	$(2, 5)$	1	$(2, 5)$	$(39, 120)$
$\mathbf{t}_3$	$(4, 5)$	0	$(0, 0)$	$(0, 0)$
$\mathbf{t}_4$	$(4, 15)$	0	$(0, 0)$	$(0, 0)$

$\mathcal{C}_2$	

eliminate non-skyline tuples

$$\mathcal{C}_1 \text{ and } \mathcal{C}_2 \text{ use SOR with } V \text{ to make } E_{pk}(S(\mathbf{t}_{min})) = E_{pk}(127)$$

Party $\mathcal{C}_1$					
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V	$(\mathbf{t}_i[1]', \mathbf{t}_i[2]')$	$(\mathbf{p}_i[1]', \mathbf{p}_i[2]')$	$S(\mathbf{t}_i)$
$\mathbf{t}_1$	$(1, 15)$	0	$(0, 0)$	$(0, 0)$	67
$\mathbf{t}_2$	$(2, 5)$	1	$(2, 5)$	$(39, 120)$	127
$\mathbf{t}_3$	$(4, 5)$	0	$(0, 0)$	$(0, 0)$	37
$\mathbf{t}_4$	$(4, 15)$	0	$(0, 0)$	$(0, 0)$	76

$\mathcal{C}_2$	

Secure Skyline Protocol

eliminate non-skyline tuples

secure dominance protocol

Party $\mathcal{C}_1$							$\mathcal{C}_2$	
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V	$(\mathbf{t}_i[1]', \mathbf{t}_i[2]')$	$(\mathbf{p}_i[1]', \mathbf{p}_i[2]')$	$S(\mathbf{t}_i)$	V		
$\mathbf{t}_1$	$(1, 15)$	0	$(0, 0)$	$(0, 0)$	67	0		
$\mathbf{t}_2$	$(2, 5)$	1	$(2, 5)$	$(39, 120)$	127	0		
$\mathbf{t}_3$	$(4, 5)$	0	$(0, 0)$	$(0, 0)$	37	1		
$\mathbf{t}_4$	$(4, 15)$	0	$(0, 0)$	$(0, 0)$	76	1		

Secure Skyline Protocol

eliminate non-skyline tuples

$$\textit{make } E_{pk}(S(\mathbf{t}_i)) = E_{pk}(127), \textit{ where } \mathbf{t}_i \textit{ is dominated by } \mathbf{t}_{min}$$

Party $\mathcal{C}_1$							
$\mathbf{t}_i$	$(\mathbf{t}_i[1], \mathbf{t}_i[2])$	V	$(\mathbf{t}_i[1]', \mathbf{t}_i[2]')$	$(\mathbf{p}_i[1]', \mathbf{p}_i[2]')$	$S(\mathbf{t}_i)$	V	$S(\mathbf{t}_i)$
$\mathbf{t}_1$	$(1, 15)$	0	$(0, 0)$	$(0, 0)$	67	0	67
$\mathbf{t}_2$	$(2, 5)$	1	$(2, 5)$	$(39, 120)$	127	0	127
$\mathbf{t}_3$	$(4, 5)$	0	$(0, 0)$	$(0, 0)$	37	1	127
$\mathbf{t}_4$	$(4, 15)$	0	$(0, 0)$	$(0, 0)$	76	1	127

$\mathcal{C}_2$	

Outline

- Problem setting
- Paillier crypto scheme
- Basic primitive subprotocols
- Secure dominance protocol
- Secure skyline protocol
- Experimental results

Experiment Setup

Protocols:

- BSSP: Basic Secure Skyline Protocol
- FSSP: Fully Secure Skyline Protocol

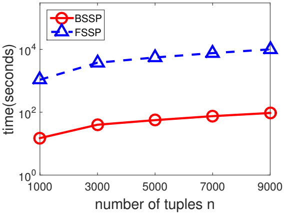
Datasets:

- NBA: real NBA dataset
- INDE: independent dataset
- CORR: correlated dataset
- ANTI: anti-correlated dataset

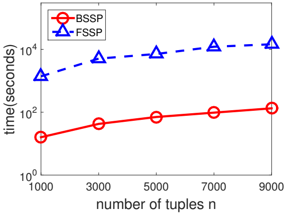
Goal:

- evaluate the performance and scalability of our protocols

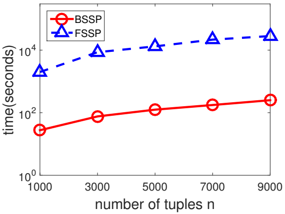
The impact of n (m=2, K=512)



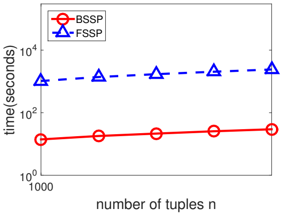
(a) time cost of CORR



(b) time cost of INDE

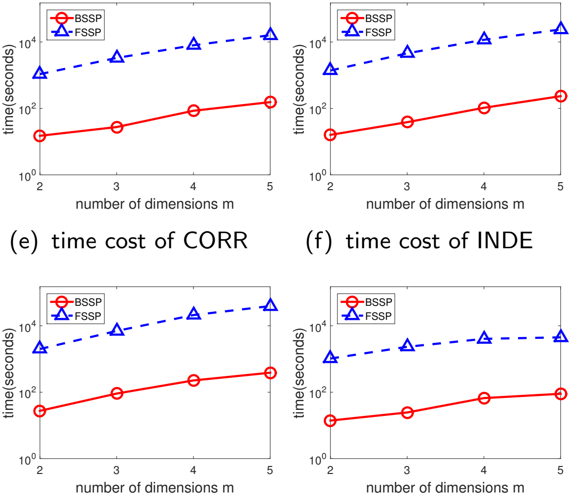


(c) time cost of ANTI



(d) time cost of NBA

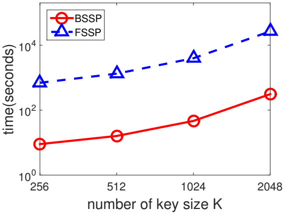
The impact of m (n=1000, K=512)



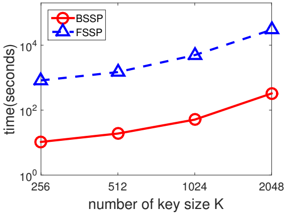
(e) time cost of CORR (f) time cost of INDE

(g) time cost of ANTI (h) time cost of NBA

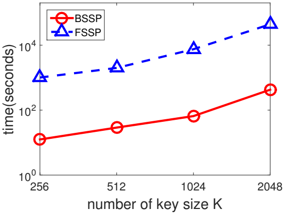
The impact of K (n=1000, m=2)



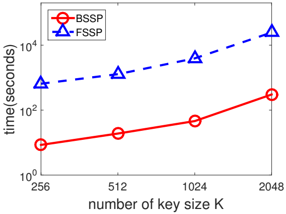
(i) time cost of CORR



(j) time cost of INDE



(k) time cost of ANTI



(l) time cost of NBA

Conclusion and Future Work

Conclusion

- Proposed a secure dominance sub-protocol.
- Proposed a fully secure skyline protocol.
- Demonstrated practical using simulation.

Future work

- Further optimization of algorithm complexity and running time.

Thank You!!!