NDSS SYMPOSIUM

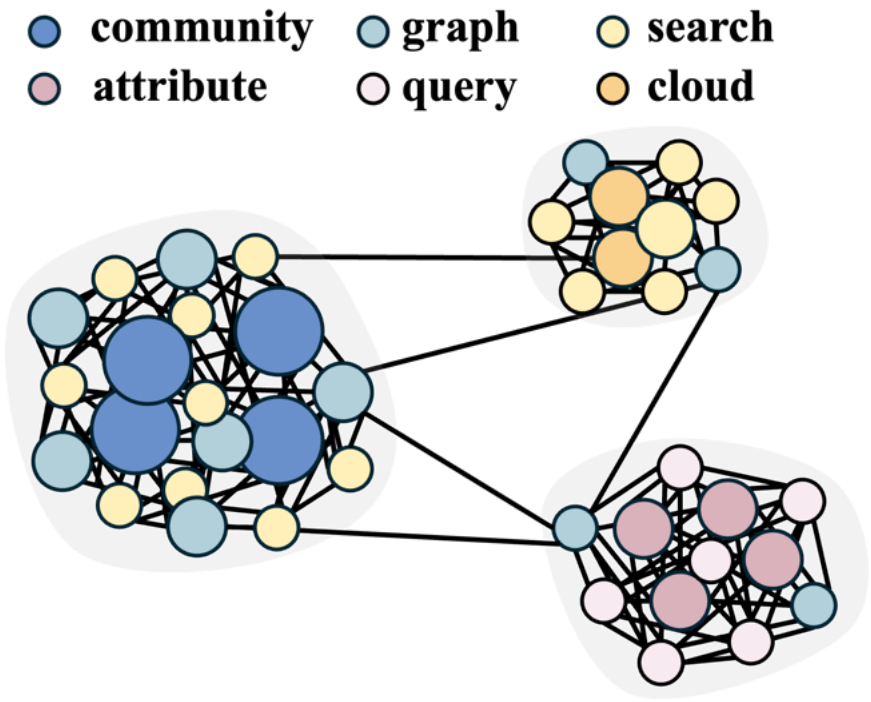
PACS: **P**rivacy-Preserving **A**ttribute-Driven **C**ommunity **S**earch over Attributed Graphs

Fangyuan Sun[1], Yaxi Yang[2], Jia Yu[1] , Jianying Zhou[2]

1 QINGDAO UNIVERSITY 青島大学

2 SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

BACKGROUND



- **Social Networks**

  Marketing and Content Delivery

- **Citation Networks**

  Group Discovery

- **Biology Networks**

  Functional Module Identification

CONTRIBUTION

- **Secure Community Index for Attribute Privacy**

  Asymmetric inner-product-based encrypted attribute evaluation with no leakage

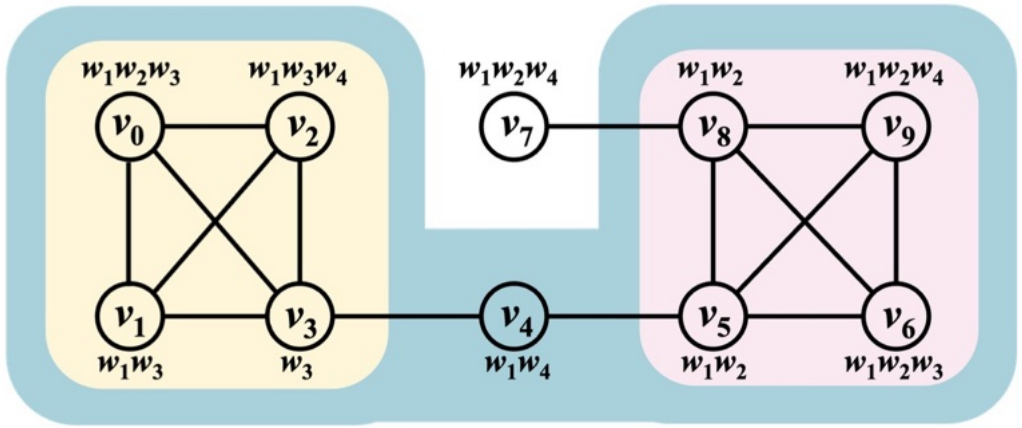
- **Secure Edge Table for Community Privacy**

  Homomorphic edge retrieval with full community confidentiality

- **Provably Secure and Efficient Attribute-Driven Community Search**

  CQA2-security, and efficient community search

DEFINITION



Community Quality

- Structural Cohesiveness

  minimum degree constraint: *k*-core

- Attributes Correlated(with the search attributes)

the semantic similarity of search attributes with community attributes
: attribute-driven score

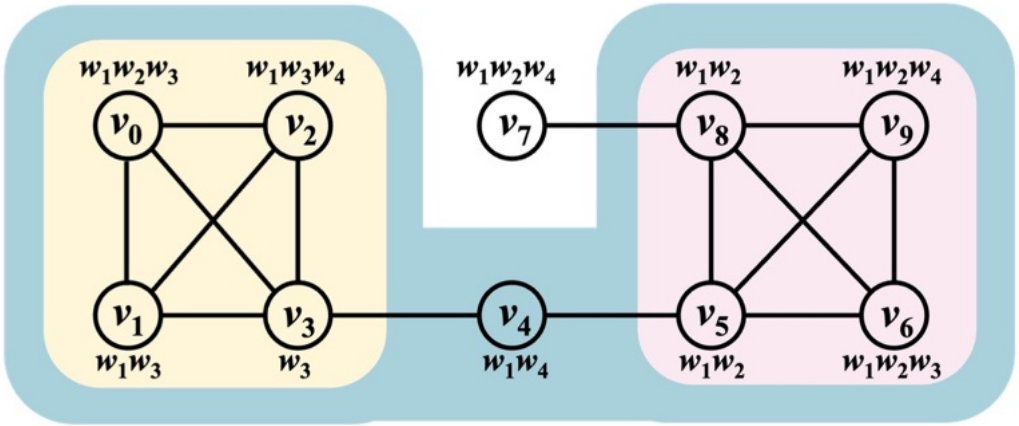
DEFINITION

Attribute-Driven Community Search

Attributed graph $G = (V, E, W)$, query attributes $W' \subseteq W$, threshold $\theta > 0$

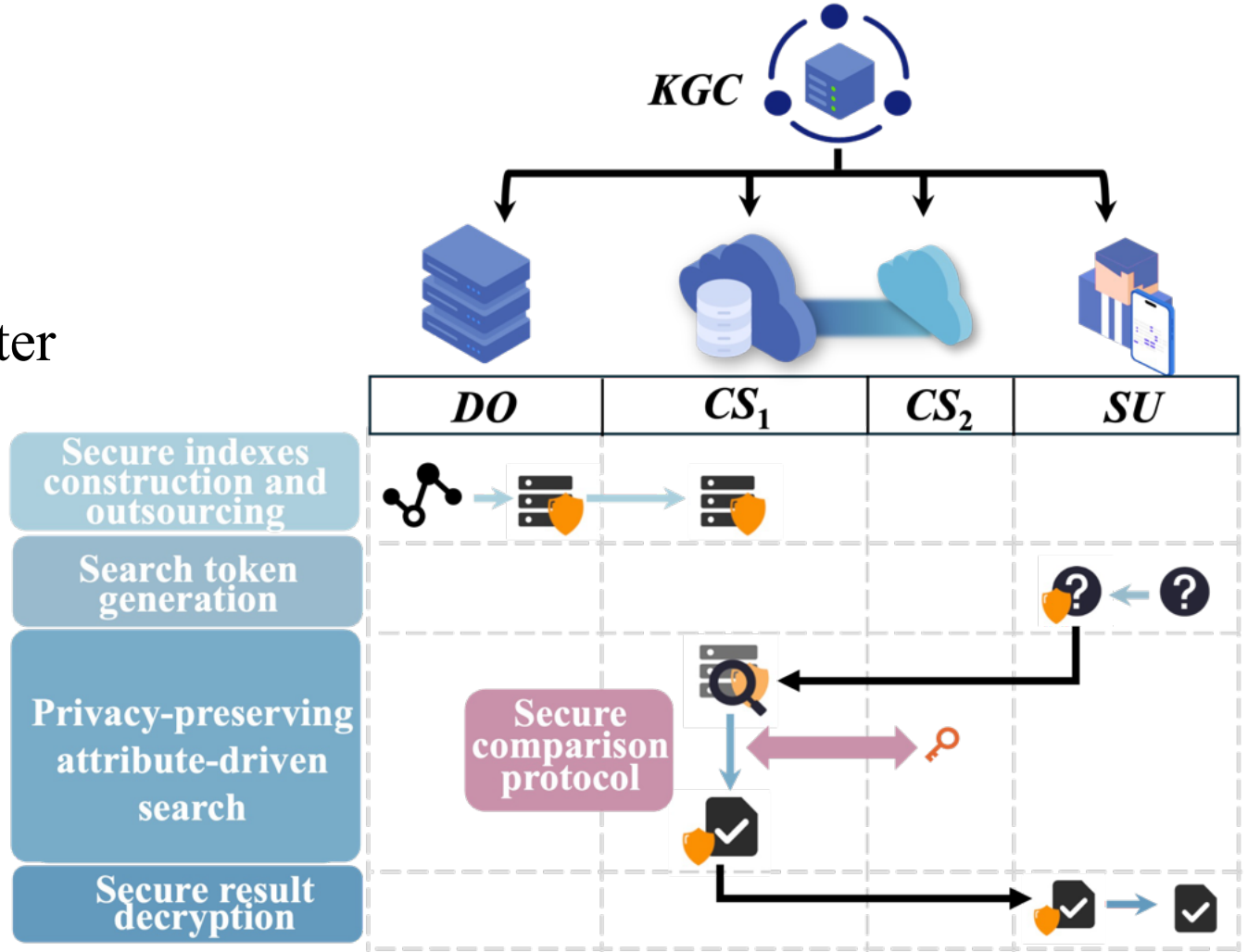
Community C satisfying:

- **Structural Cohesiveness:** Core number of $C \geq \theta$
- **Attribute Correlation:** Attribute-driven score $f(C, W') > 0$ and maximized among all communities meeting the structural constraint

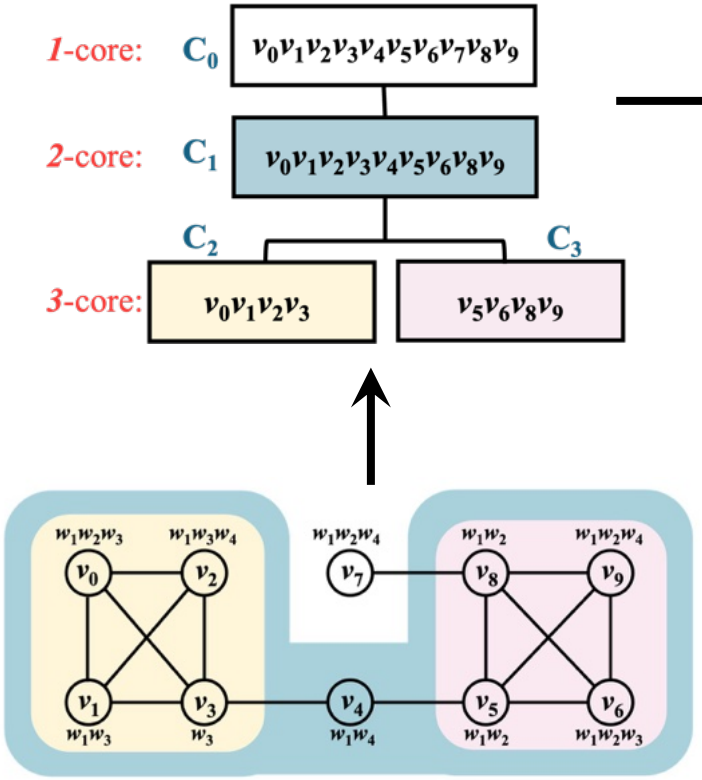


SYSTEM MODEL

- **KGC**: Key Generation Center
- **DO**: Data Owner
- **CS**: Cloud Servers
- **SU**: Search Users



DATA STRUCTURE



➢ Community Index

id	core	$\vec{a}$	$\vec{e}$
C_2	3	[2.25 0.25 4.00 0.25]	[0 0 1 1 0 1 0 1 0 0 1 0 0 1 0]
G	1	[8.10 3.60 2.50 0.40]	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
C_1	2	[7.11 2.78 2.78 1.00]	[1 1 1 1 1 1 0 1 1 1 1 1 1 1 1]
C_3	3	[4.00 4.00 0.25 0.50]	[1 1 0 0 1 0 0 0 0 1 0 1 1 0 0]

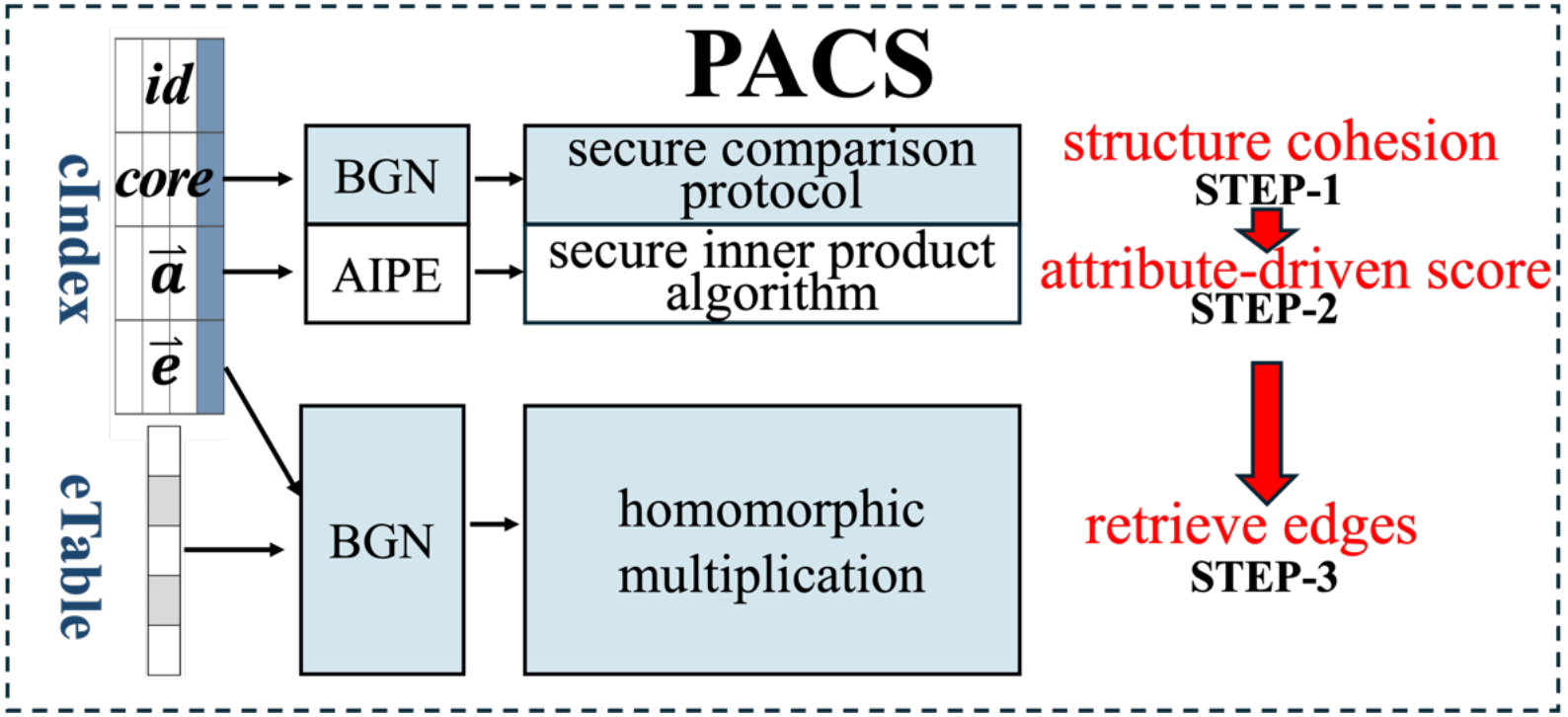
$$
\begin{aligned}
f(C,S) &= \sum_{i=0}^{t-1} \frac{(a_i s_i)^2}{z} \\
&= \sum_{i=0}^{t-1} \frac{(a_i s_i)^2}{z} \\
&= \frac{(a_0 s_0)^2 + (a_1 s_1)^2 + \ldots + (a_{t-1} s_{t-1})^2}{z} \\
&= \frac{{a_0}^2 s_0 + {a_1}^2 s_1 + \ldots + {a_{t-1}}^2 s_{t-1}}{z} \qquad //\vec{s} \in [0,1]^t \\
&= \frac{{a_0}^2}{z} s_0 + \frac{{a_1}^2}{z} s_1 + \ldots + \frac{{a_{t-1}}^2}{z} s_{t-1} \\
&= [\frac{{a_0}^2}{z}, \frac{{a_1}^2}{z}, \ldots, \frac{{a_{t-1}}^2}{z}] \cdot [s_0, s_1, \ldots, s_{t-1}] \\
&= \overrightarrow{(a^2/z)} \cdot \vec{s},
\end{aligned}
$$

where $\cdot$ represents the inner product operation.

➢ Edge Table

$v_5 \| \| v_9$	$v_8 \| \| v_9$	$v_0 \| \| v_3$	$v_1 \| \| v_3$	$v_6 \| \| v_9$	$v_1 \| \| v_2$	$v_7 \| \| v_8$	$v_0 \| \| v_1$	$v_4 \| \| v_5$	$v_5 \| \| v_6$	$v_0 \| \| v_2$	$v_5 \| \| v_8$	$v_6 \| \| v_8$	$v_2 \| \| v_3$	$v_3 \| \| v_4$

FRAMEWORK


PACS
cIndex
id
core
$\vec{a}$
$\vec{e}$
eTable
BGN
AIPE
BGN
secure comparison protocol
secure inner product algorithm
homomorphic multiplication
structure cohesion
STEP-1
attribute-driven score
STEP-2
retrieve edges
STEP-3

OVERVIEW

P A C S

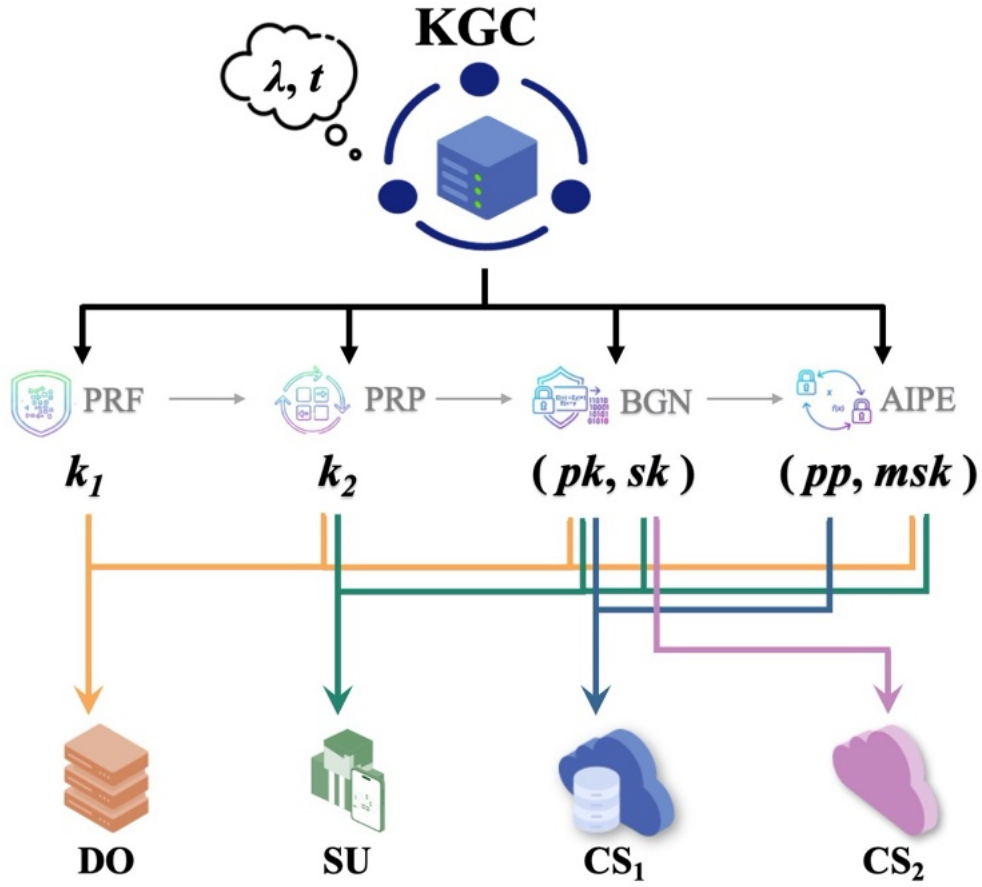
- **KeyGen (λ, t):** Generates key set K using security parameter λ and vector length t
- **EncIndex (K, *cIndex*, *eTable*):** Generates secure community index CI and secure edge table ET from community index and edge table
- **GenToken (K, $\vec{s}$, θ):** Generates search token *Token* from K, search attribute vector $\vec{s}$, and threshold θ
- **Search (CI, ET, *Token*):** Performs privacy-preserving attribute-driven community search and outputs encrypted edge table R
- **Decrypt(K, R):** Obtains the edge information *edge* of target community using K and R

PACS: KeyGen

Algorithm 1: KeyGen

Input: A security parameter λ and the vector length t;
Output: A key set K;

1 $k_1 \xleftarrow{\$} \{0,1\}^{\lambda}$;
2 $k_2 \xleftarrow{\$} \{0,1\}^{\lambda}$;
3 $(pk, sk) \leftarrow$BGN.KeyGen(λ);
4 $(pp, msk) \leftarrow$AIPE.KeyGen(λ, t);
5 **return** $K = \{k_1, k_2, (pk, sk), (pp, msk)\}$.



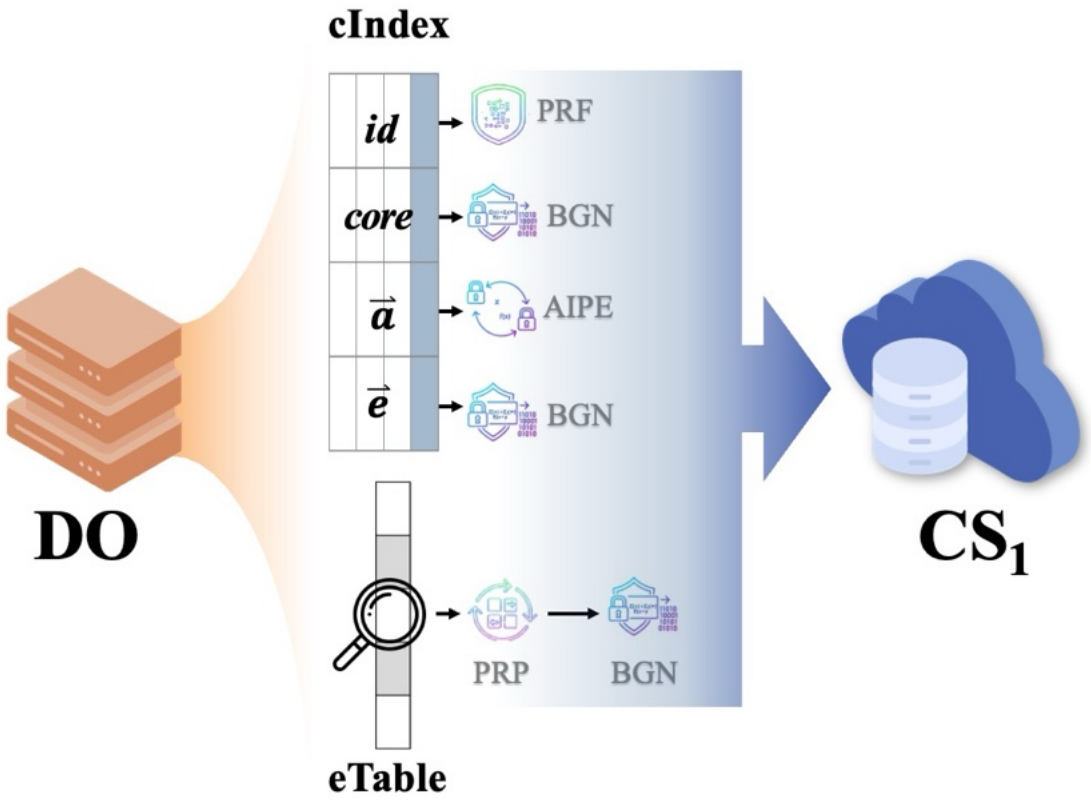
PACS: EncIndex

Algorithm 2: EncIndex

Input: A key set K, a community index $cIndex$ and an edge table $eTable$.

Output: A secure community index CI and a secure edge table ET.

parse K as $\{k_1, k_2, pk, msk\}$;
initialize two empty dictionaries CI and ET;
// encrypt $cIndex$
for *each community in* $cIndex$ **do**
  $id' \leftarrow \text{PRF}(k_1, id)$;
  $[\![core]\!] \leftarrow \text{BGN.Enc}(pk, core)$;
  $[\vec{a}] \leftarrow \text{AIPE.EncP}(msk, \vec{a})$;
  $[\![\vec{e}]\!] \leftarrow \text{BGN.Enc}(pk, \vec{e})$;
  $(id', [\![core]\!], [\vec{a}], [\![\vec{e}]\!])$ is randomly assigned to a row in CI;
// encrypt $eTable$
for *each edge* $u||v$ *in* $eTable$ **do**
  $(u||v)' \leftarrow PRP(k_2, u||v)$;
  $[\![u||v]\!] \leftarrow \text{BGN.Enc}(pk, (u||v)')$;
  add $[\![u||v]\!]$ into ET;
return (CI, ET).

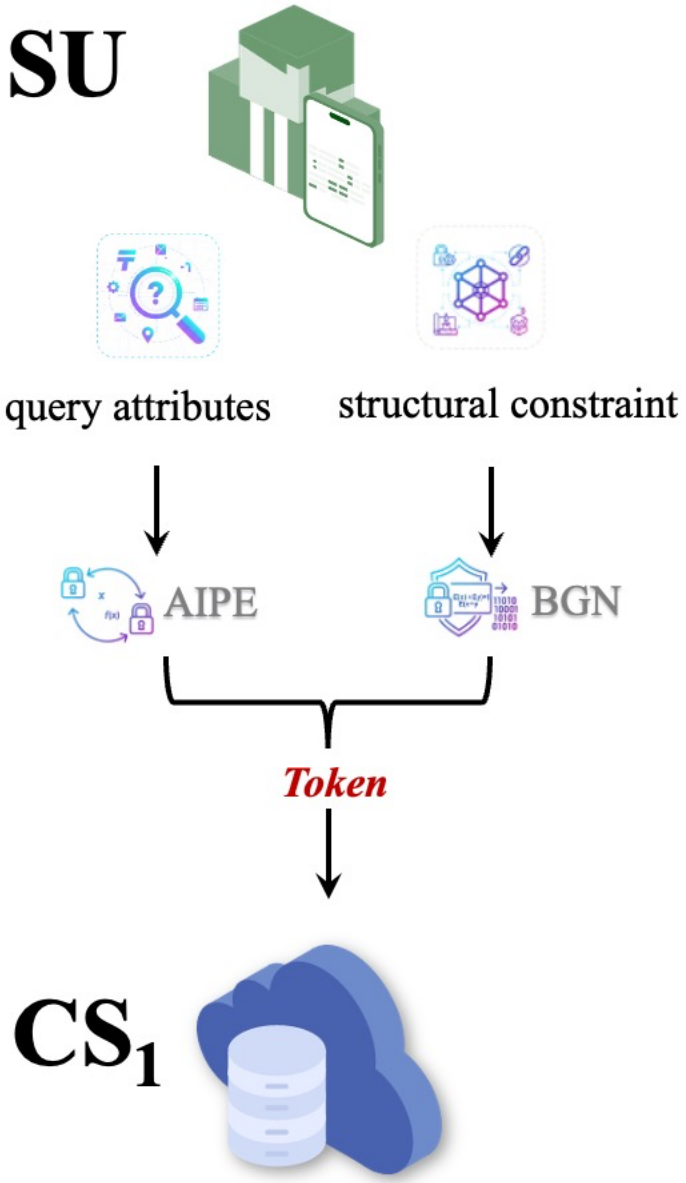


PACS: GenToken

Algorithm 3: GenToken

Input: A key set K, a search attribute vector $\vec{s}$ and an integer θ.
Output: A search token $Token$.

1 parse K as $\{pk, msk\}$;
2 $[\vec{s}] \leftarrow$ AIPE.EncQ$(msk, \vec{s})$;
3 $[\![\theta]\!] \leftarrow$BGN.Enc$(pk, \theta)$;
4 **return** $Token = \{[\vec{s}], [\![\theta]\!]\}$.

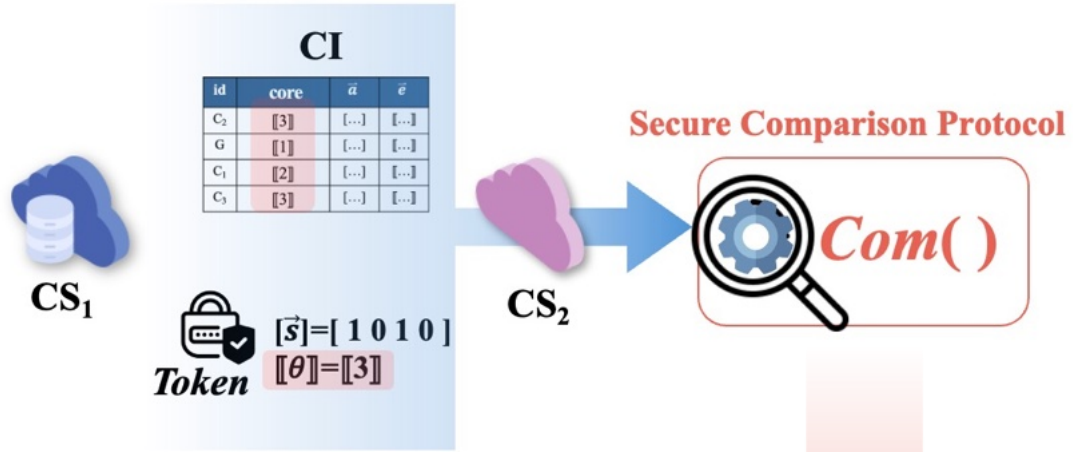


PACS: Search

Algorithm 4: Search

Input: A secure community index CI, a secure edge table ET and a search token $Token$.
Output: An encrypted table R.

parse $Token$ as $\{[\vec{s}], [\![\theta]\!]\}$;
initialize an empty table R of length n;
// STEP-1: Core Number Filtering
initialize an empty dictionary F;
for i *from* 0 *to* δ **do**
  if $Com(CI[i].[\![core]\!], [\![\theta]\!])$ **then**
    add $CI[i]$ to F;
// STEP-2: Attribute-Driven Score Calculation
initialize an edge vector $[\vec{g}]$;
for *each community* C *in* F **do**
  $f(C, [\vec{s}]) \leftarrow$AIPE.IP($pp$, $[\vec{s}]$, $C.[\vec{a}]$);
$[\vec{g}] \leftarrow C.[\vec{e}]$, where C is the community with the highest f;
// STEP-3: Result Return
for i *from* 0 *to* $n-1$ **do**
  $R[i] \leftarrow e([\vec{g}]_i, ET[i])$;
return R.



Protocol 1: Com($[\![m_1]\!], [\![m_2]\!]$)

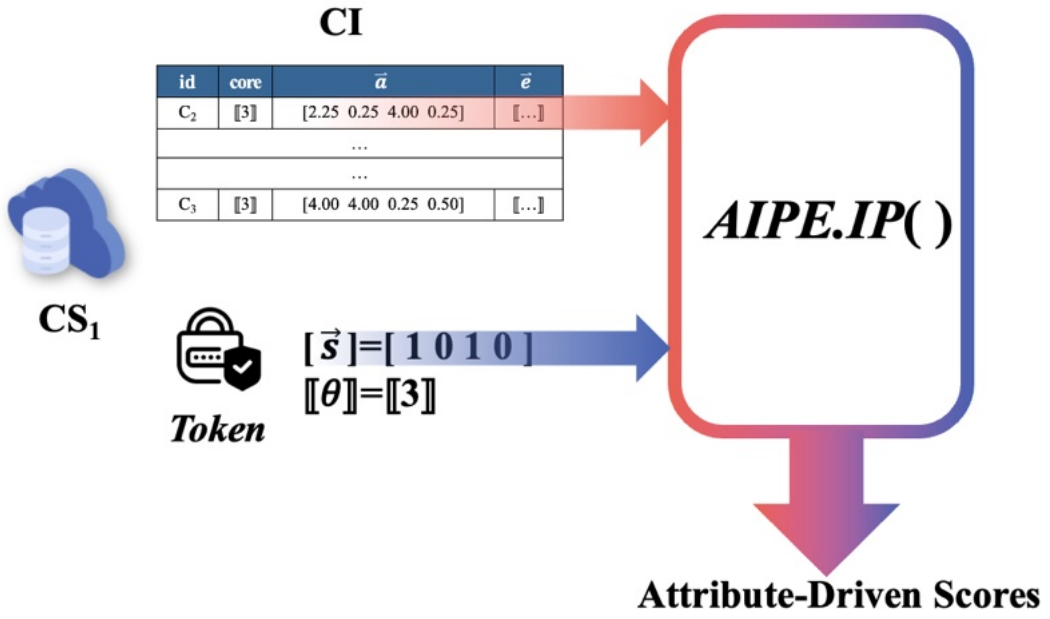
// CS_1
$[\![m_1']\!] \leftarrow [\![m_1]\!]^2 \cdot [\![1]\!]$;
$[\![m_2']\!] \leftarrow [\![m_2]\!]$;
select $b \in \{0,1\}$ and $r \xleftarrow{R} \mathbb{Z}_N$;
if b==1 then
  $[\![l]\!] \leftarrow ([\![m_1']\!] \cdot [\![m_2']\!]^{-1})^r$;
else
  $[\![l]\!] \leftarrow ([\![m_1']\!]^{-1} \cdot [\![m_2']\!])^r$;
send $[\![l]\!]$ to CS_2
// CS_2
$l \leftarrow$BGN.Enc(pk, $[\![l]\!]$);
if $l > T$ then
  $f' \leftarrow 1$;
else
  $f' \leftarrow 0$;
send f' to CS_1;
// CS_1
if b==1 then
  $f \leftarrow f'$;
else
  $f \leftarrow 1 - f'$;

PACS: Search

Algorithm 4: Search

Input: A secure community index CI, a secure edge table ET and a search token $Token$.
Output: An encrypted table R.

parse $Token$ as $\{[\vec{s}], [\![\theta]\!]\}$;
initialize an empty table R of length n;
// STEP-1: Core Number Filtering
initialize an empty dictionary F;
for i *from* 0 *to* δ **do**
  if $Com(CI[i].[\![core]\!], [\![\theta]\!])$ **then**
    add $CI[i]$ to F;
// STEP-2: Attribute-Driven Score Calculation
initialize an edge vector $[\vec{g}]$;
for *each community* C *in* F **do**
  $f(C, [\vec{s}]) \leftarrow$AIPE.IP($pp$, $[\vec{s}]$, $C.[\vec{a}]$);
$[\vec{g}] \leftarrow C.[\vec{e}]$, where C is the community with the highest f;
// STEP-3: Result Return
for i *from* 0 *to* $n-1$ **do**
  $R[i] \leftarrow e([\vec{g}]_i, ET[i])$;
return R.

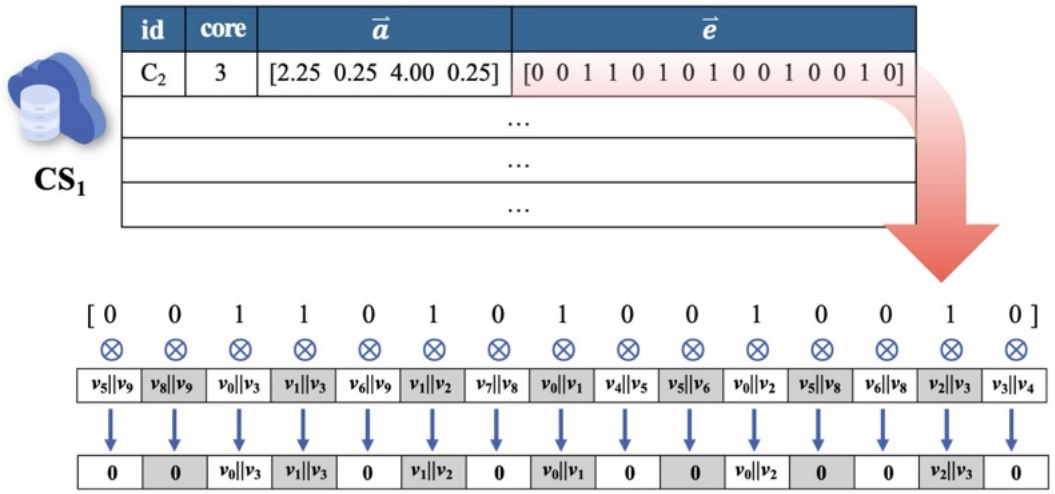


PACS: Search

Algorithm 4: Search

Input: A secure community index CI, a secure edge table ET and a search token $Token$.
Output: An encrypted table R.

parse $Token$ as $\{[\vec{s}], [\![\theta]\!]\}$;
initialize an empty table R of length n;
// STEP-1: Core Number Filtering
initialize an empty dictionary F;
for i *from* 0 *to* δ **do**
  if $Com(CI[i].[\![core]\!], [\![\theta]\!])$ **then**
    add $CI[i]$ to F;
// STEP-2: Attribute-Driven Score Calculation
initialize an edge vector $[\vec{g}]$;
for *each community* C *in* F **do**
  $f(C, [\vec{s}]) \leftarrow$AIPE.IP($pp$, $[\vec{s}]$, $C.[\vec{a}]$);
$[\vec{g}] \leftarrow C.[\vec{e}]$, where C is the community with the highest f;
// STEP-3: Result Return
for i *from* 0 *to* $n-1$ **do**
  $R[i] \leftarrow e([\vec{g}]_i, ET[i])$;
return R.

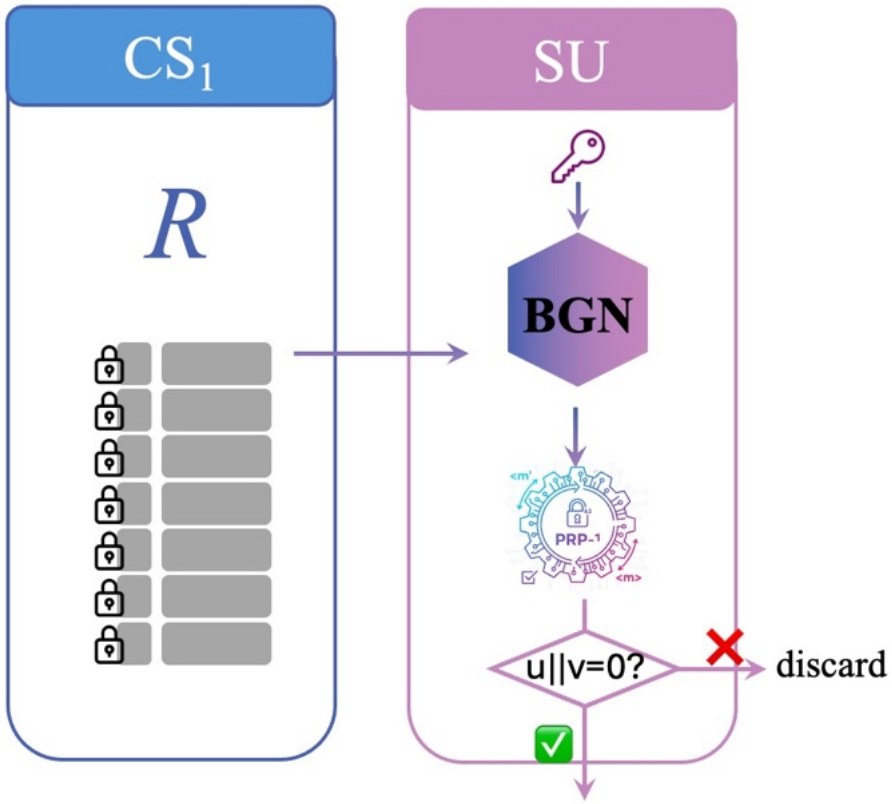


PACS: Decrypt

Algorithm 5: Decrypt

Input: A secret key set K and an encrypted edge table R.
Output: An edge table $edge$.

```
parse K as {k_2, sk};
initialize an empty table edge;
for i form 0 to n − 1 do
    (u||v)' ←BGN.Dec(sk, R[i]);
    u||v ←PRP^{-1}(k_2, (u||v)');
    if u||v≠ 0 then
        add u||v into edge;
return edge.
```



SECURITY ANALYSIS

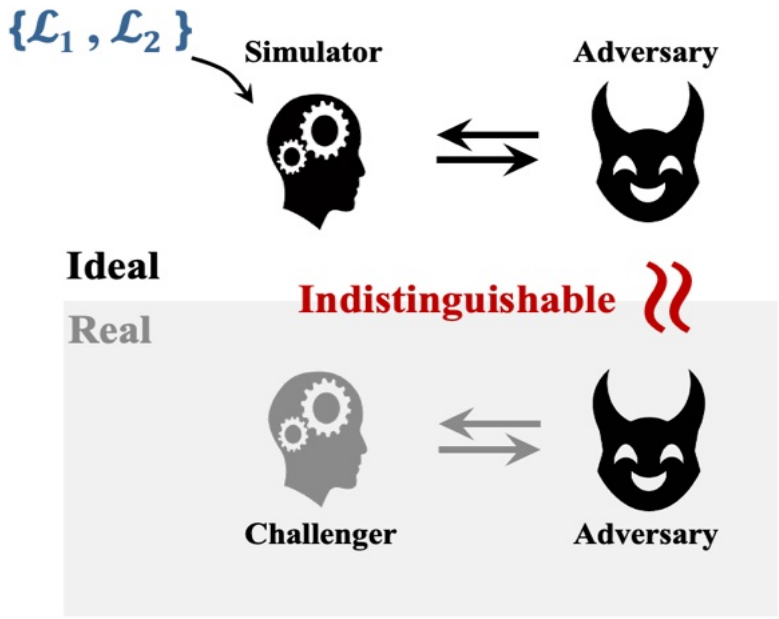
$\mathcal{L}_1$: Captures information from secure indexes

- ***n***: number of edges / length of ET
- ***t***: number of attributes
- ***δ***: number of communities / length of CI

$\mathcal{L}_2$: Captures information revealed during search

1. **Query pattern leakage**
   - ***info*$_1$**:repeated index accesses
2. **Index pattern leakage**
   - ***info*$_2$**: accessed index entries
   - ***ore***: order information between core number and constraint
   - ***num***: number of candidate communities
   - ***list***: attribute-driven scores of candidates

Against Adaptive Chosen-Query Attacks (CQA2-Security):

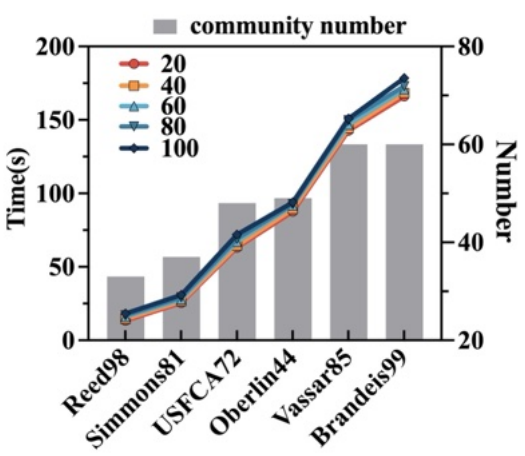


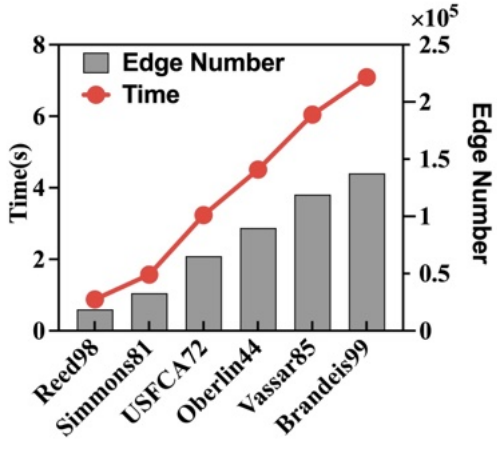
EXPERIMENT

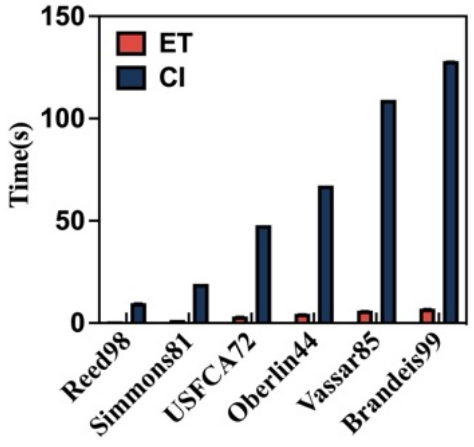
TABLE I: Dataset statistics.

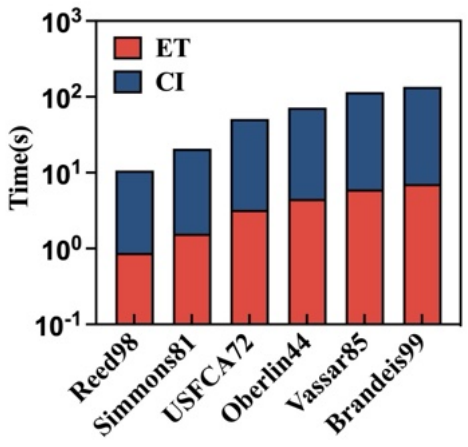
Dataset	Number of Vertices	Number of Edges	Maximum core number	Number of Communities
Reed98	962	18812	35	33
Simmons81	1519	32989	35	37
USFCA72	2683	65253	44	48
Oberlin44	2921	89913	50	49
Vassar85	3069	119162	61	60
Brandeis99	3899	137568	57	60

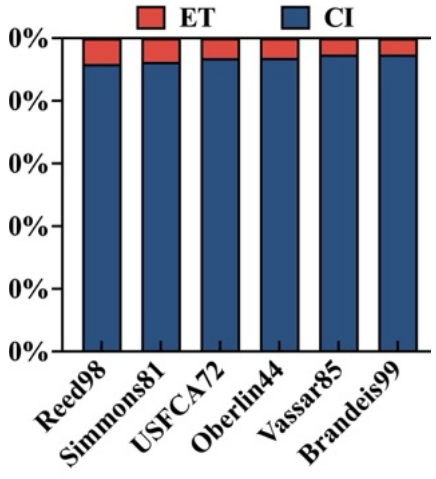
[1]https://networkrepository.com

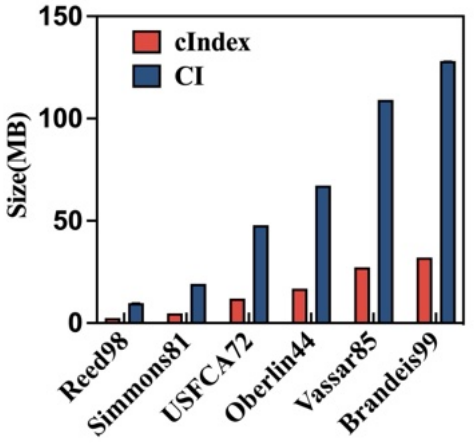


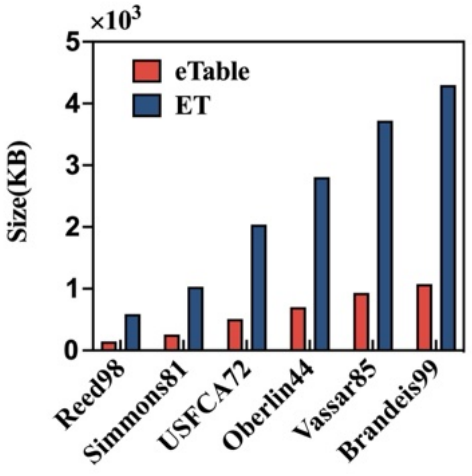




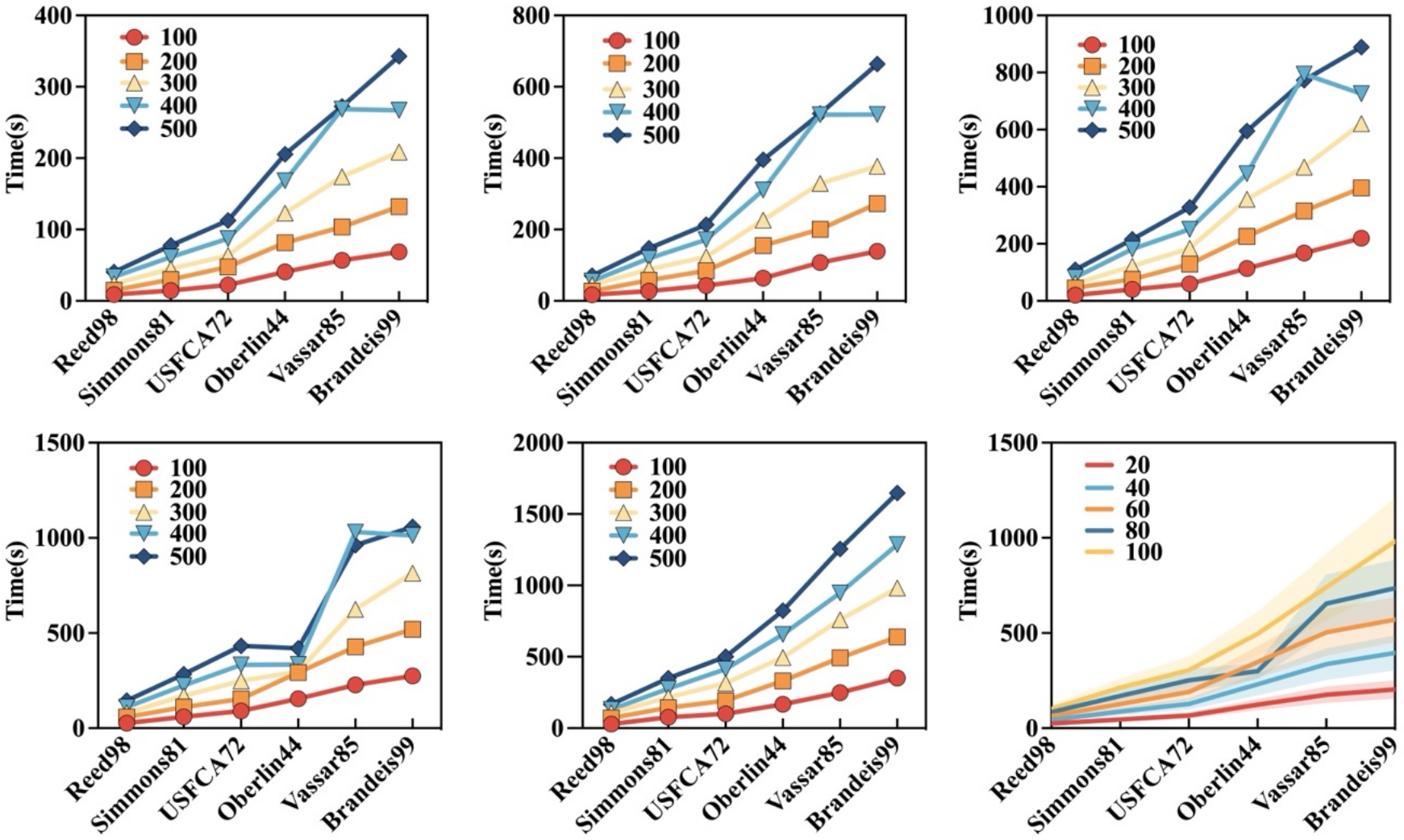




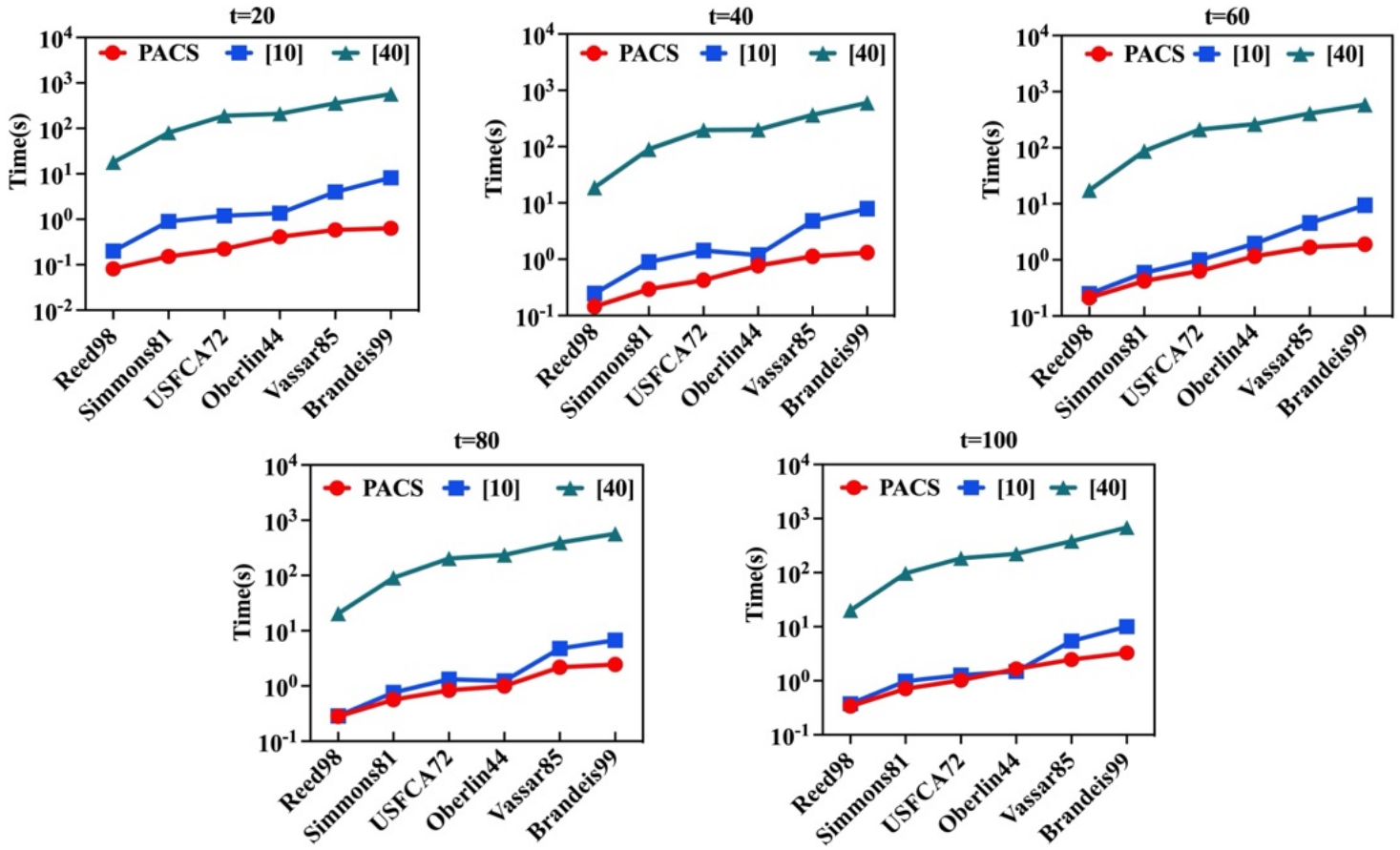




EXPERIMENT


Time(s)
100
200
300
400
500
20
40
60
80
Reed98
Simmons81
USFCA72
Oberlin44
Vassar85
Brandeis99

EXPERIMENT



[10]Y. Zhu, Q. Zhang, L. Qin, L. Chang, and J. X. Yu, “Cohesive subgraph search using keywords in large networks,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 178–191, 2020.
[40]L. Chen, C. Liu, K. Liao, J. Li, and R. Zhou, “Contextual Community Search Over Large Social Networks,” in IEEE International Conference on Data Engineering, ser. ICDE’19. IEEE, 2019, pp. 88–99.

CONCLUSION

Secure Indexes

Privacy-Preserving Search

Efficient Search

THANK YOU!

sakgofish@gmail.com