AnonPSI: An Anonymity Assessment Framework for PSI

Bo Jiang, Jian Du, Qiang Yan

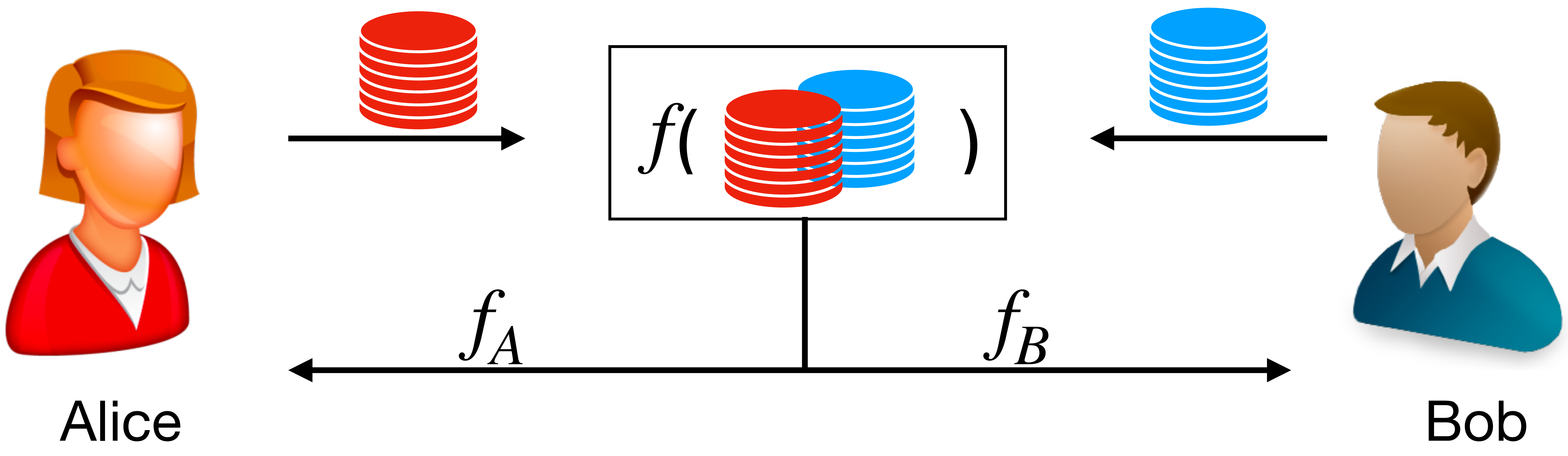
Privacy Innovation Lab
TikTok Inc.

Content

- Background
- Deterministic attack: a dynamic programming solution
- Improvement with auxiliary information
- Statistic attack: Bayesian active learning
- Experiments

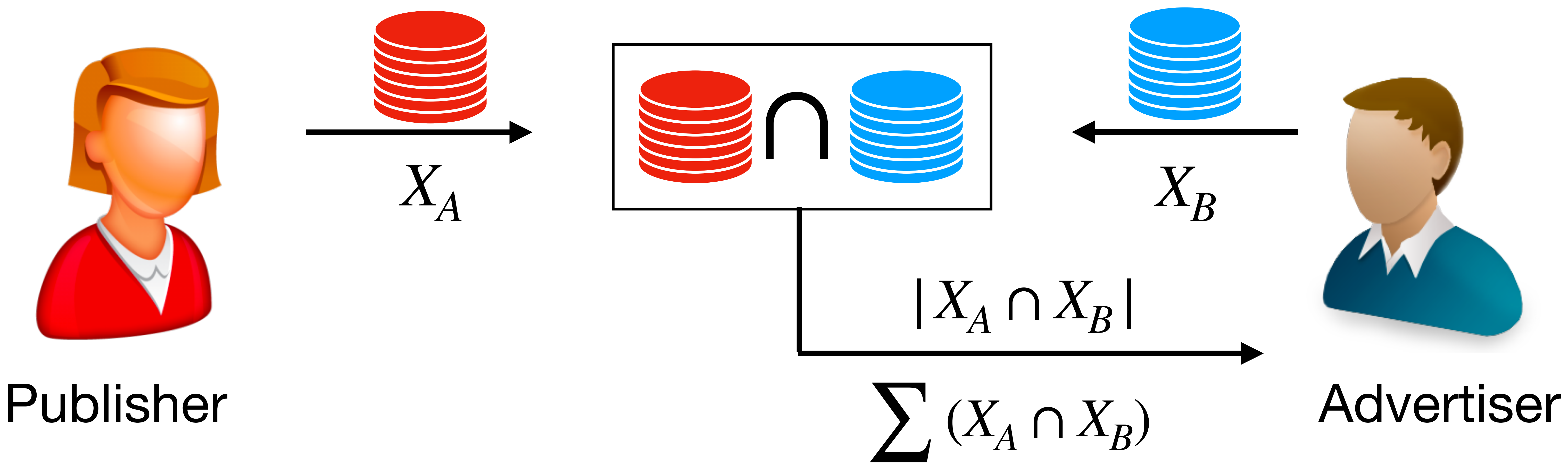
Background - Secure Two-party Computation

- Privacy-preserving: ensures that each party's input remains confidential
- Security guarantees: Provides cryptographic assurances that neither party can cheat
- Applications: financial service, healthcare, supply chain management, online voting, Ads measurement, collaborative machine learning, etc.



Background - Example of Ads measurement

- f : Ads conversion rate / revenue from intersecting converted individuals.
- X_A : user set from the publisher that viewed the ads
- X_B : user set from the advertiser that purchased the product

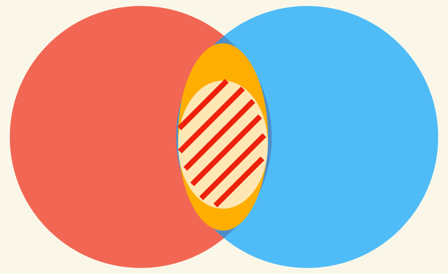
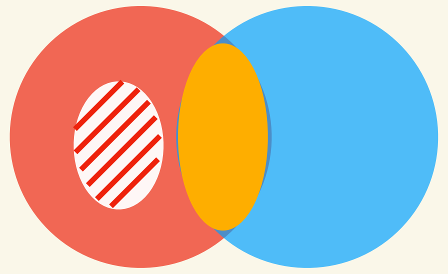
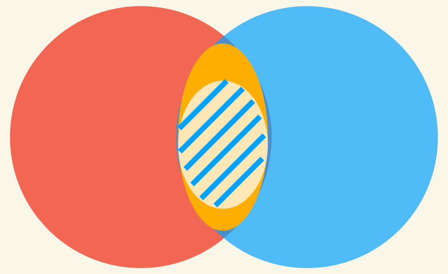
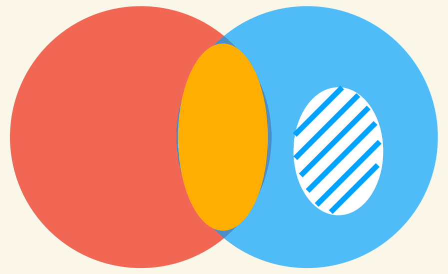


Background - privacy leakage in intersection size revealing protocols

- Intersection size revealing protocols: Protocols that only returns the cardinality of the intersection, with/without other side-information, eg. PSI-CA, PSI-SUM, etc.

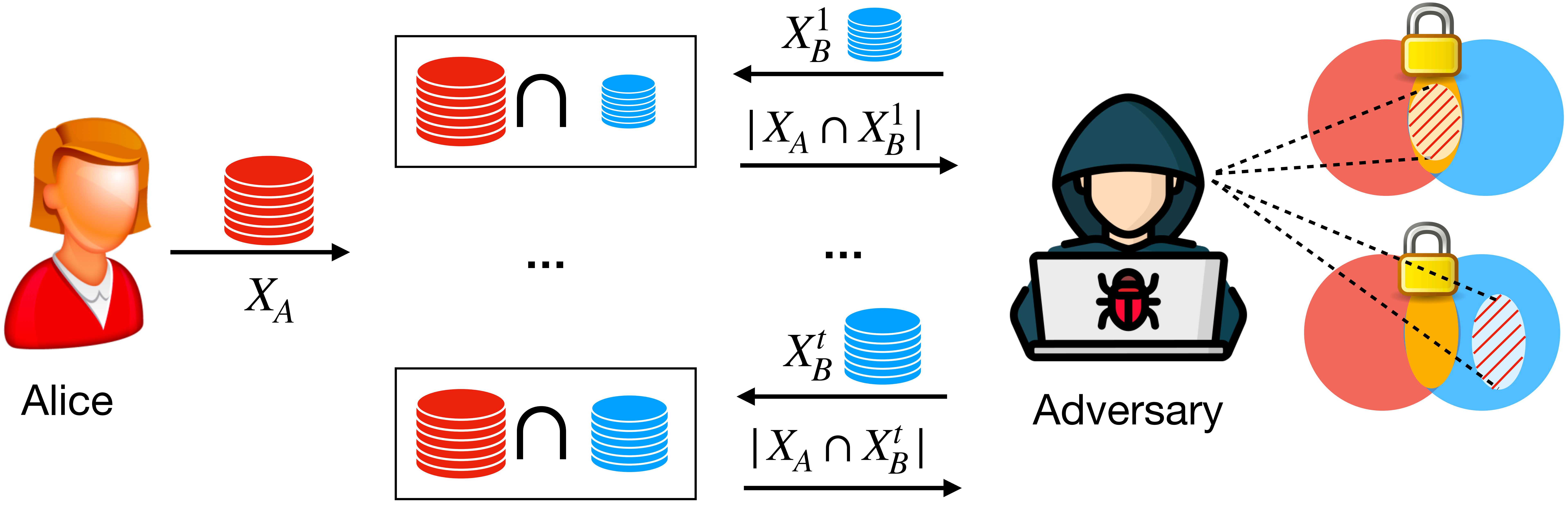
- Why hide the intersection?

All types of Leakages

	Track users at the intersect	Track users not at the intersect
Ads provider A tracks opt-out users	 A identifies users consumed at B Keep sending B's ad to them	 A identifies users not consumed at B Sell their info to B's competitors
Advertiser provider B tracks opt-out users	 B identifies costumers using A Sell their info at a higher price to A	 B identifies costumers not using A Target them through other platforms

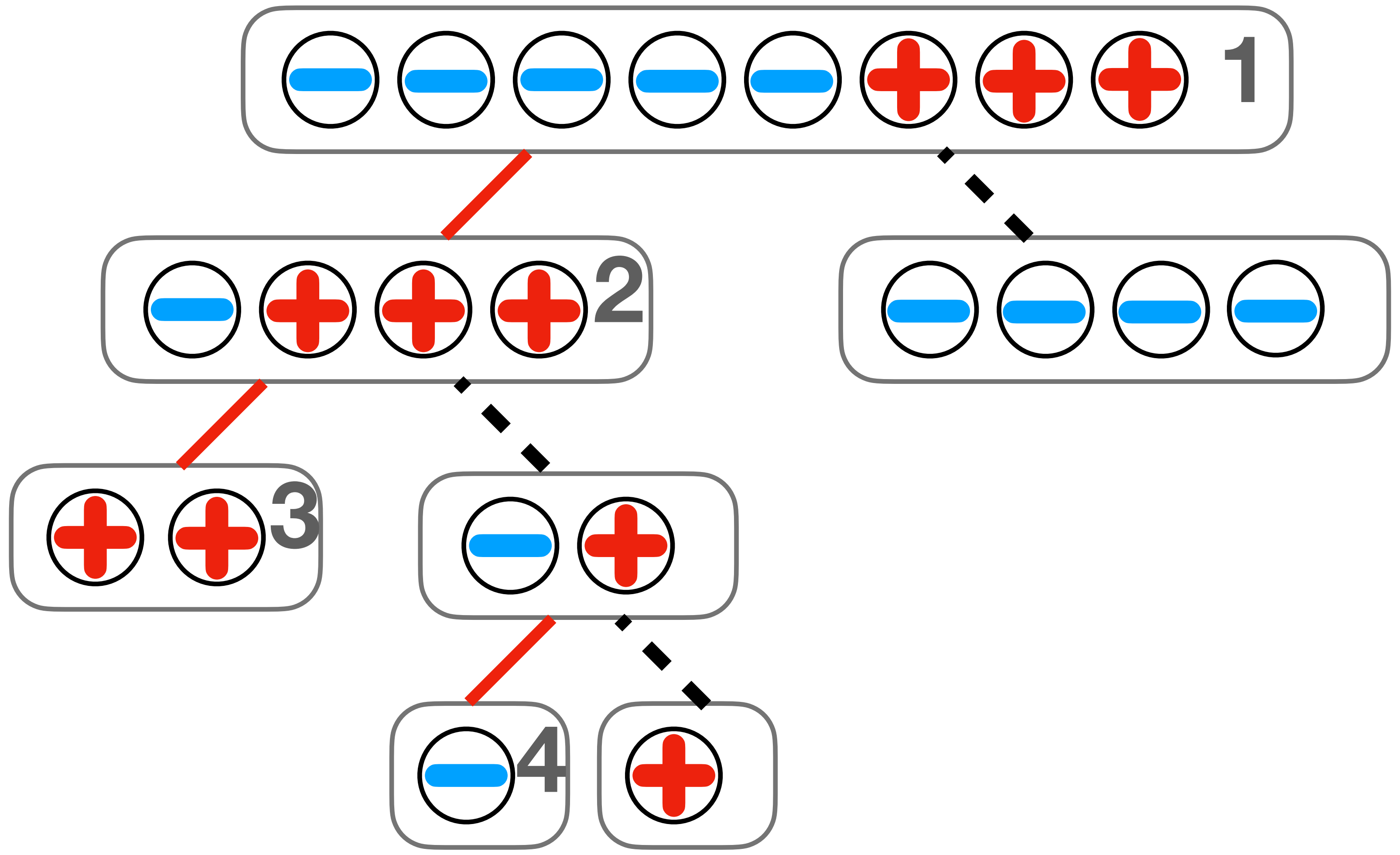
Background - set membership inference attack

- Malicious party infers a set of users' membership by invocations of a sequence of protocols
- X_B^i denotes the adversary's input subset for the i-th protocol call
- Brute force attack: adversary submits one person at a time and determines his membership

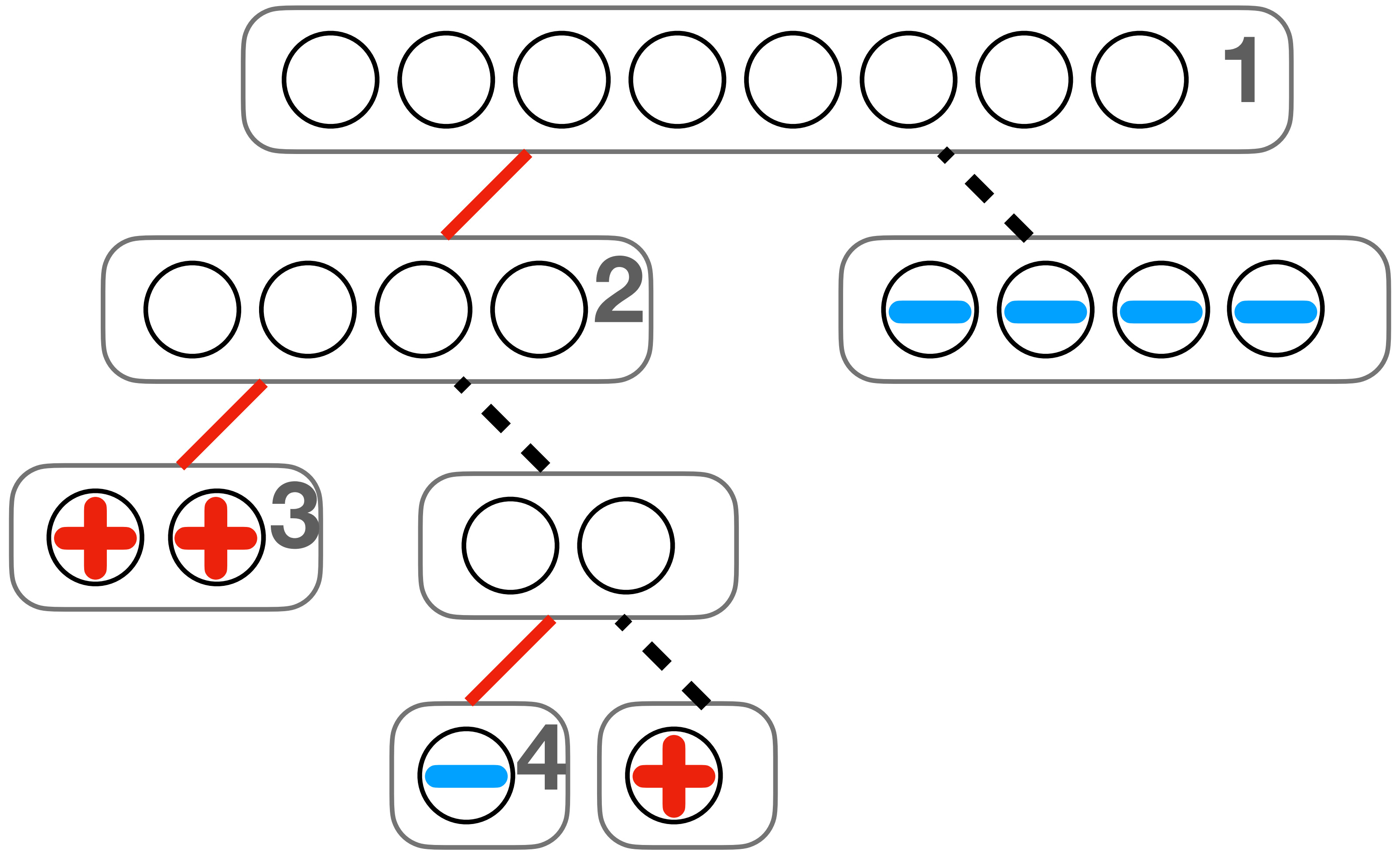


Baseline attack algorithm [1]

- Setup binary tree with each node to be a subset of users
- Visit nodes via *Priority*-based depth-first search (*Priority* = Intersection size (*IS*) /# individuals in the node)
- *IS* in the right child = *IS* in the parent - *IS* in the left child
- Classify current node if *Priority* = 0 (negative membership) or *Priority* = 1 (positive membership)



Attacking process



Attacker's point of view

[1]: Guo, Xiaojie et al. "Birds of a Feather Flock Together: How Set Bias Helps to Deanonymize You via Revealed Intersection Sizes." *USENIX Security Symposium* (2022).

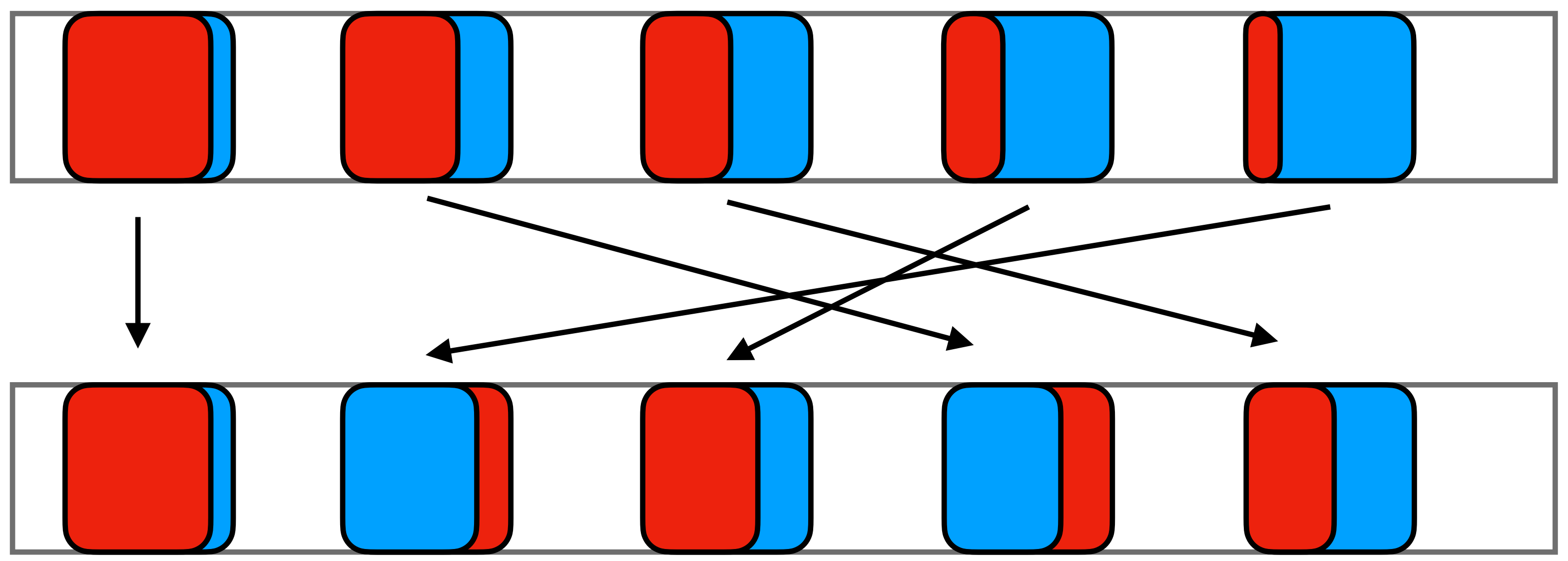
Improvements over the baseline

- Improvement 1: leverage both positive and negative membership by redefining *Priority*

Priority Queue

Baseline

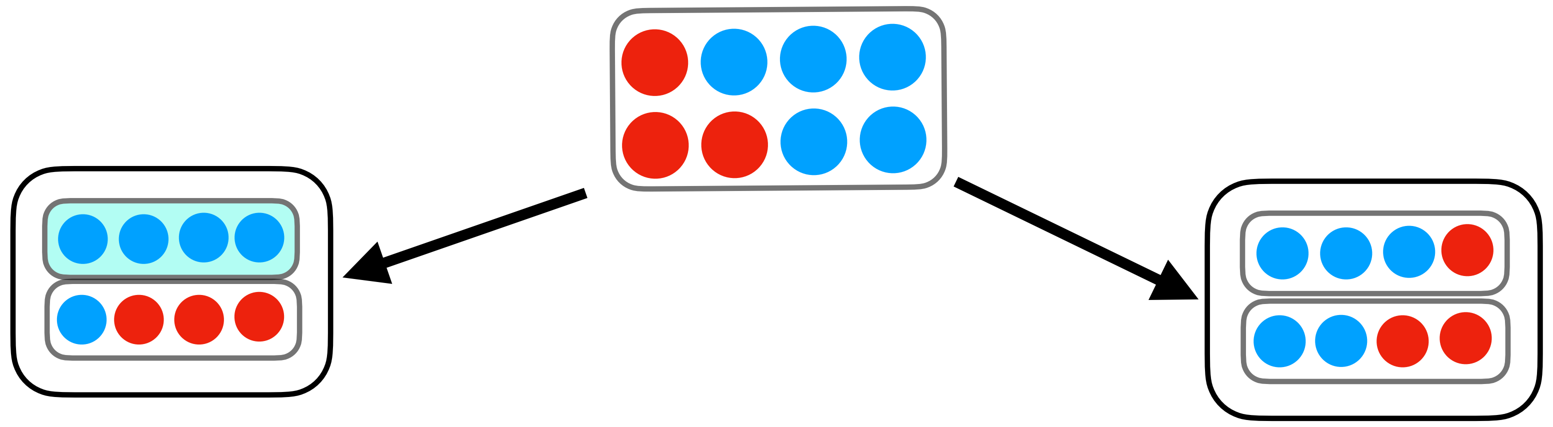
Modified Algorithm



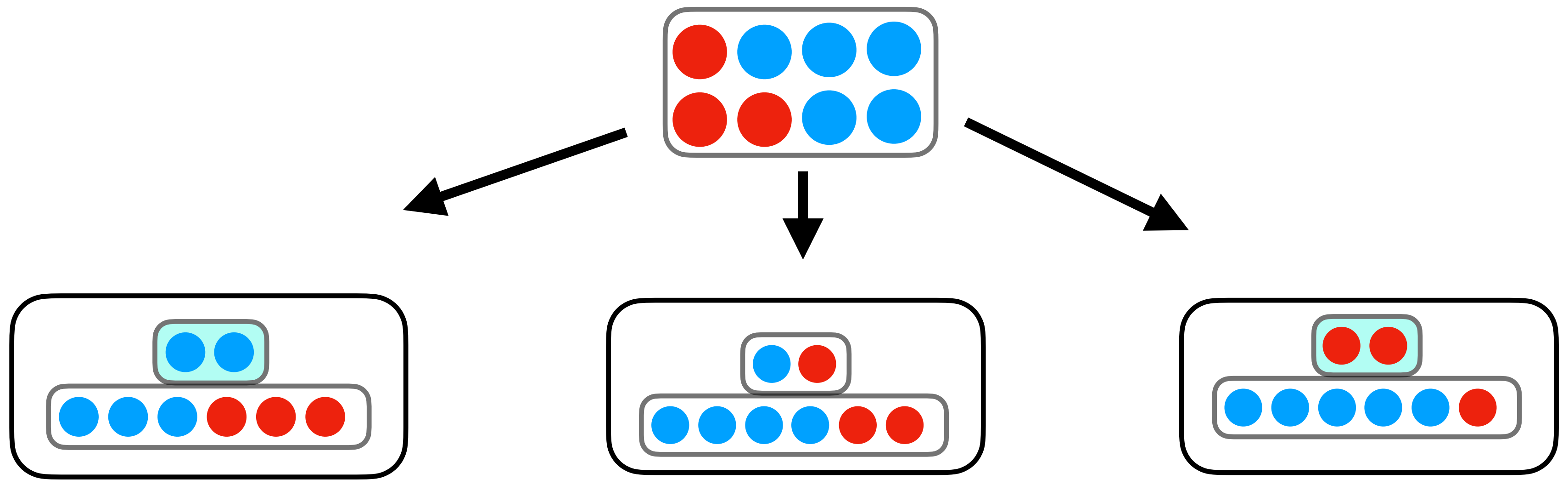
Positive members

Negative members

- Improvement 2: optimal tree partition, benefit for cases with limited protocol calls



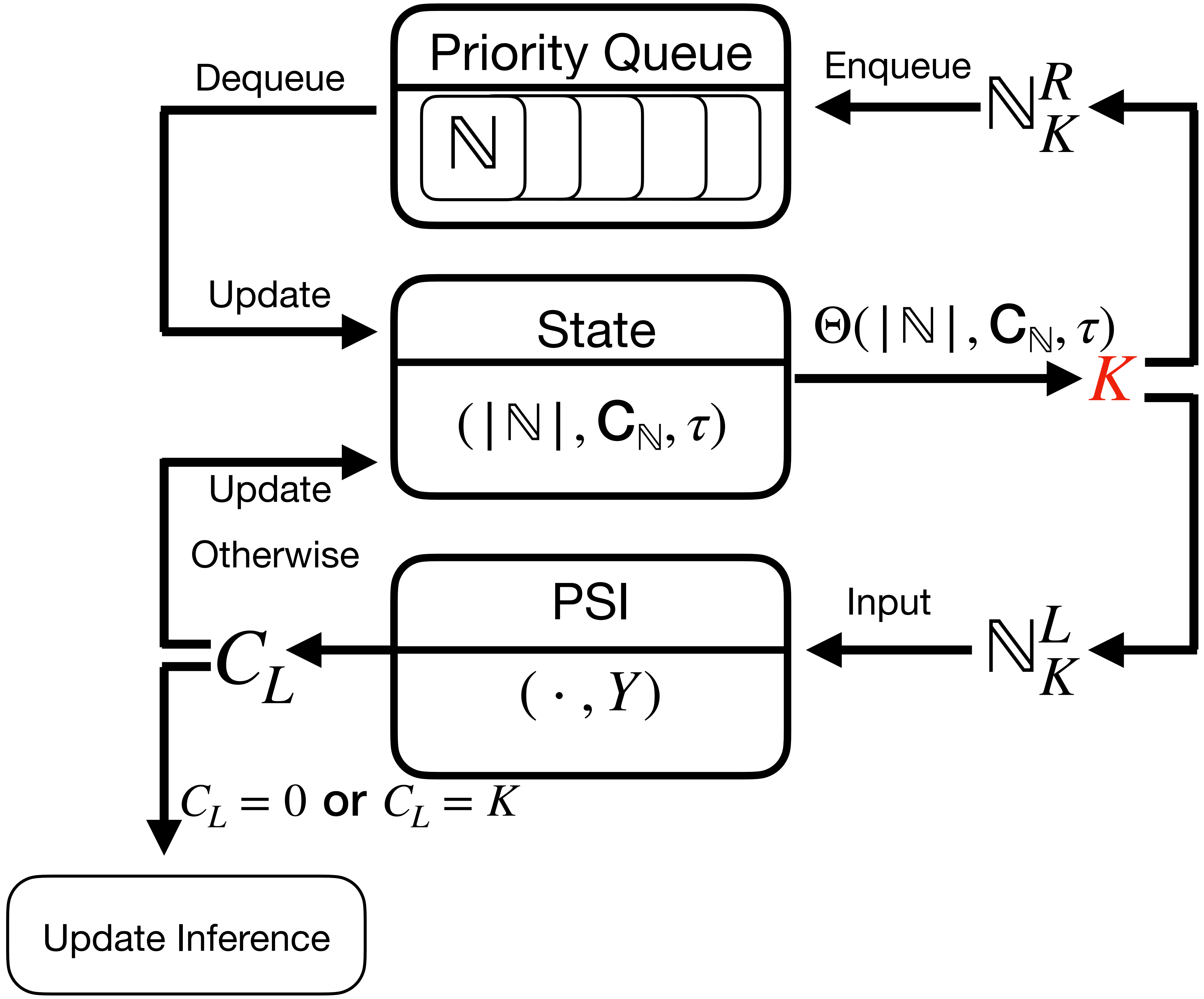
Call #	Partition	Exp.Leak
1	0	0
2	4	0.57



Call #	Partition	Exp.Leak
1	0	0
2	2	0.928

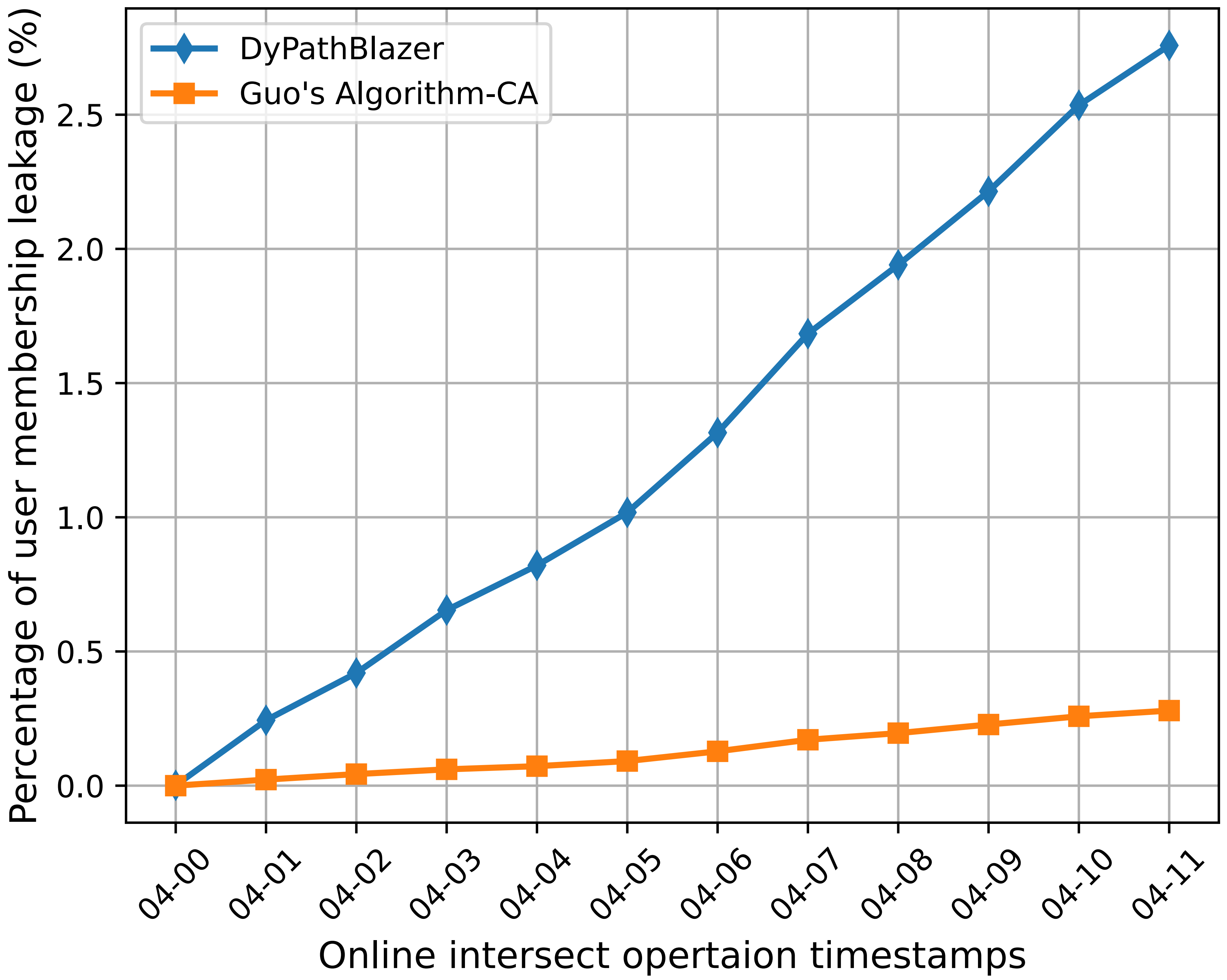
Deterministic dynamic problem approach - DyPathBlazer

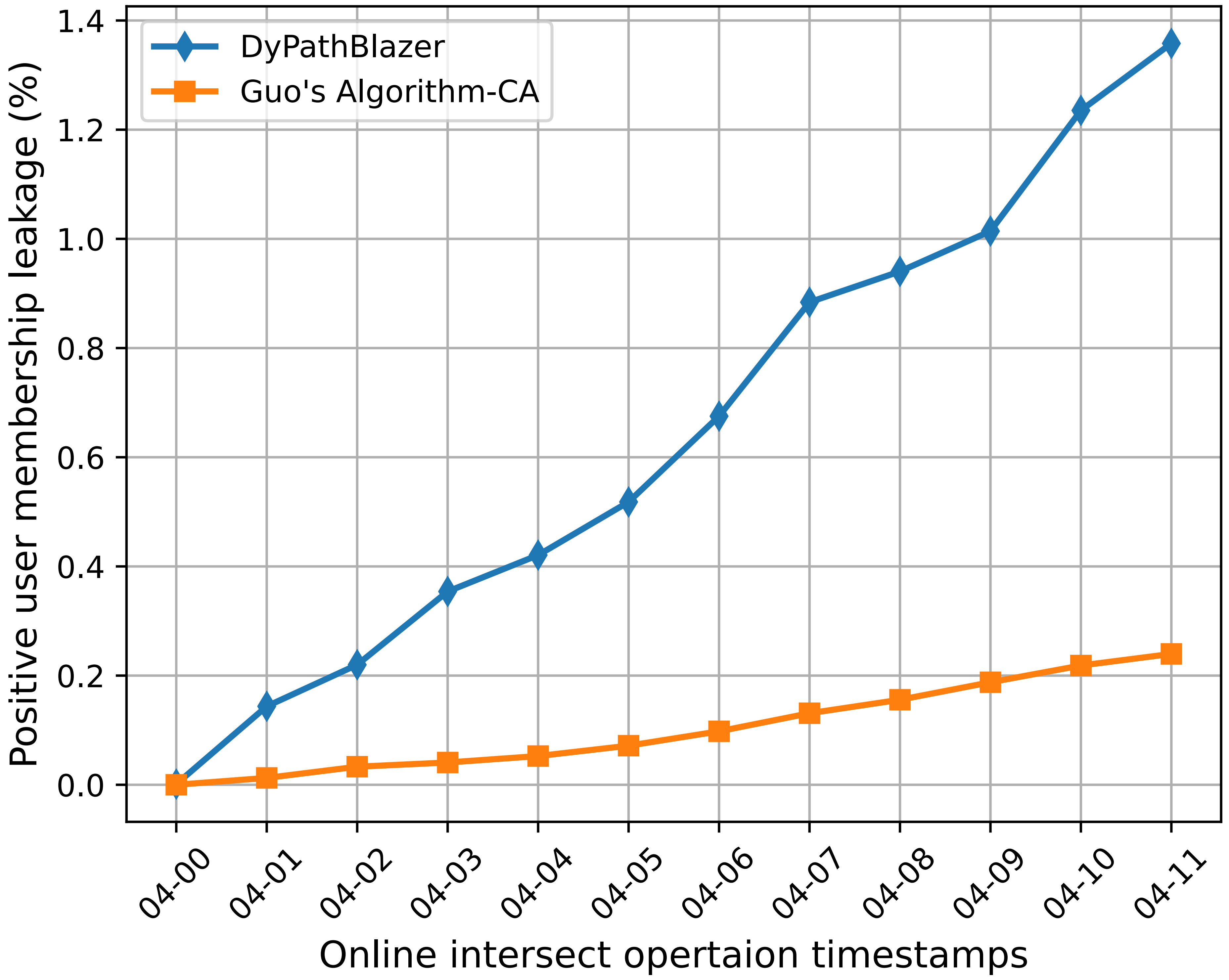
- State: $(|\mathbb{N}|, C_{\mathbb{N}}, \tau)$, number of elements in current set, number of intersected elements, protocol invocation budget.
- *Priority* $= \max(C_{\mathbb{N}}/|\mathbb{N}|,\ \ 1 - C_{\mathbb{N}}/|\mathbb{N}|)$
- $\Theta(|\mathbb{N}|, C_{\mathbb{N}}, \tau)$ stores the optimal partition factor K that maximizes the expected inferred memberships under current state
- $\Theta(|\mathbb{N}|, C_{\mathbb{N}}, \tau)$, is pre-calculated offline by back tracking (dynamic programming).



Experiments with Covid-19 tracking record*

Leakage (left): individuals who tested and the results are inferred during [04-00, 04-11]
Leakage (right):individuals who tested and the results are inferred positive during [04-00, 04-11]

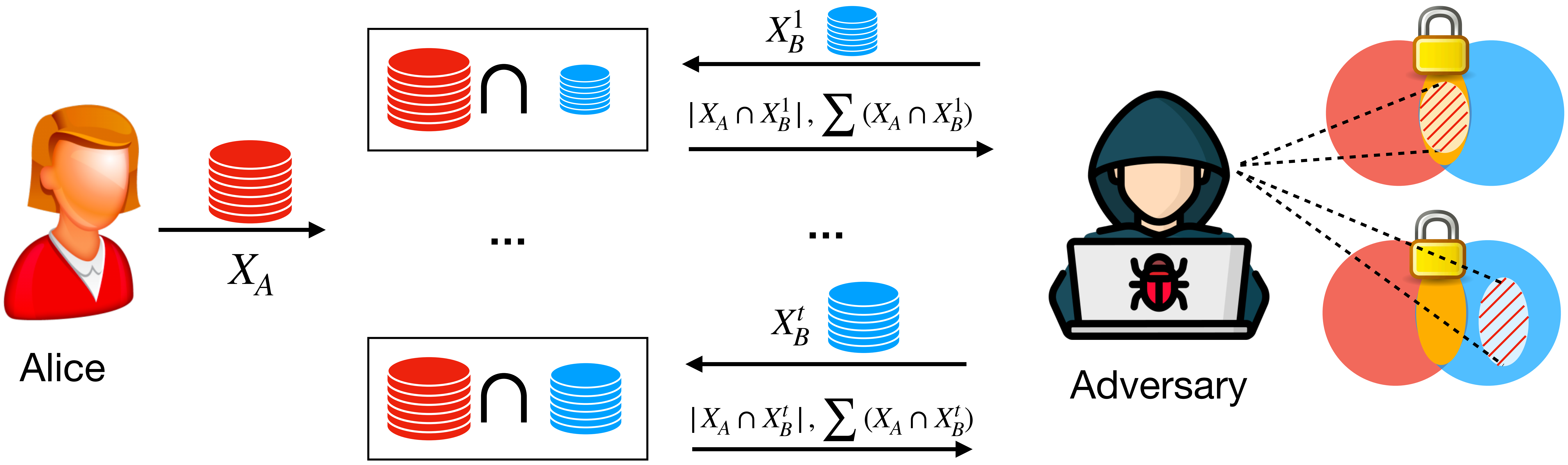




*: Machine learning-based prediction of covid-19 diagnosis based on symptoms, url: https://github.com/nshomron/covidpre

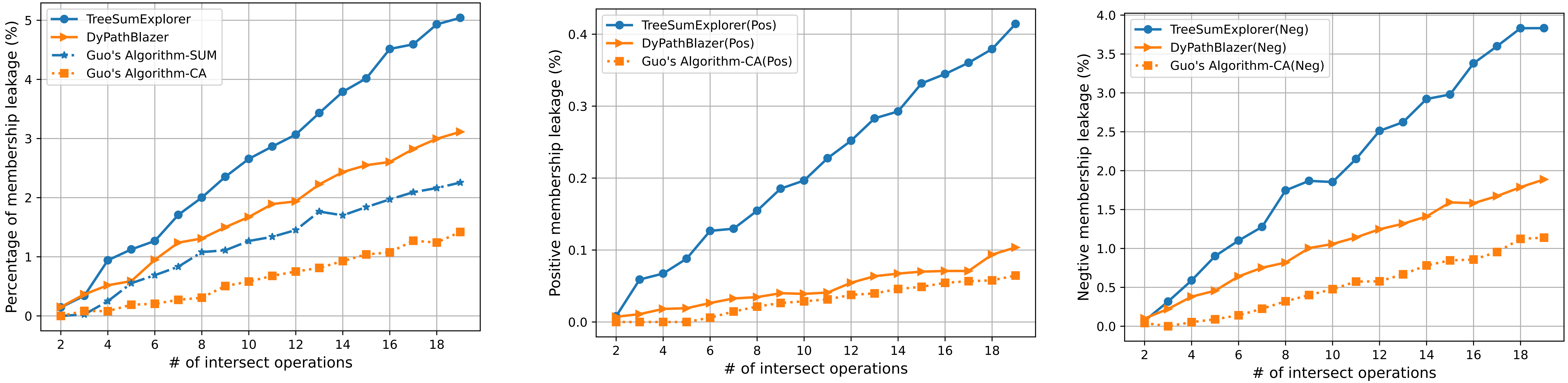
Combined with auxiliary information - TreeSumExplorer

- Observations: Count, $|X_A \cap X_B^k|$ and SUM, $\sum (X_A \cap X_B^k)$
- Solves an offline N-Sum problem (find elements in X_B^t of length "Count" that sum up to SUM)



Experiments with ads display and click dataset*

Advertising company targets the product company:



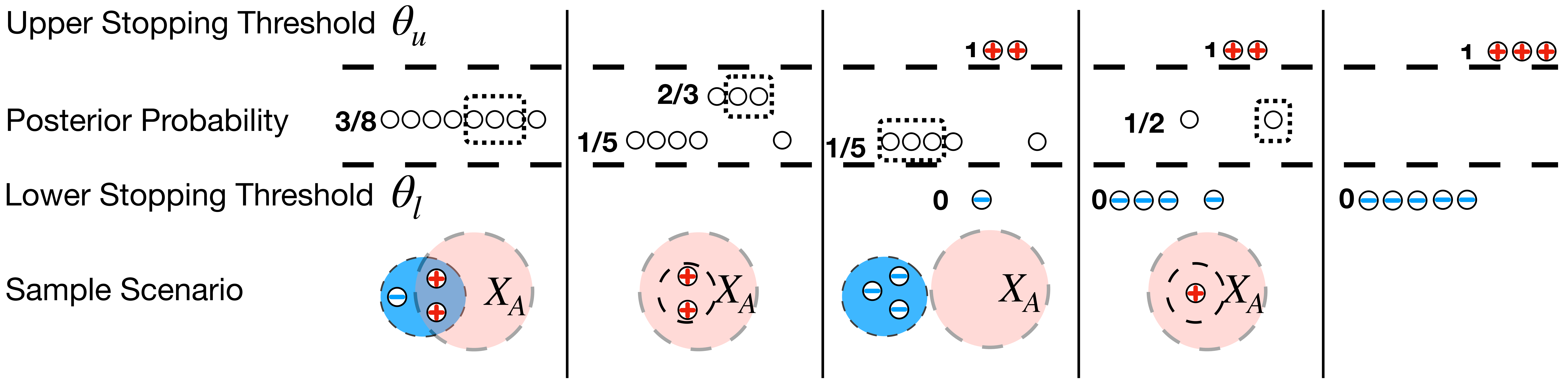
DyPathBlazer can be adapted to focus on inferring positive/negative membership only, by redefining the *Priority*

The attack efficiency improved significantly with auxiliary information

*: Taobao Display Advertisement Click-Through Rate Prediction Dataset, url: https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

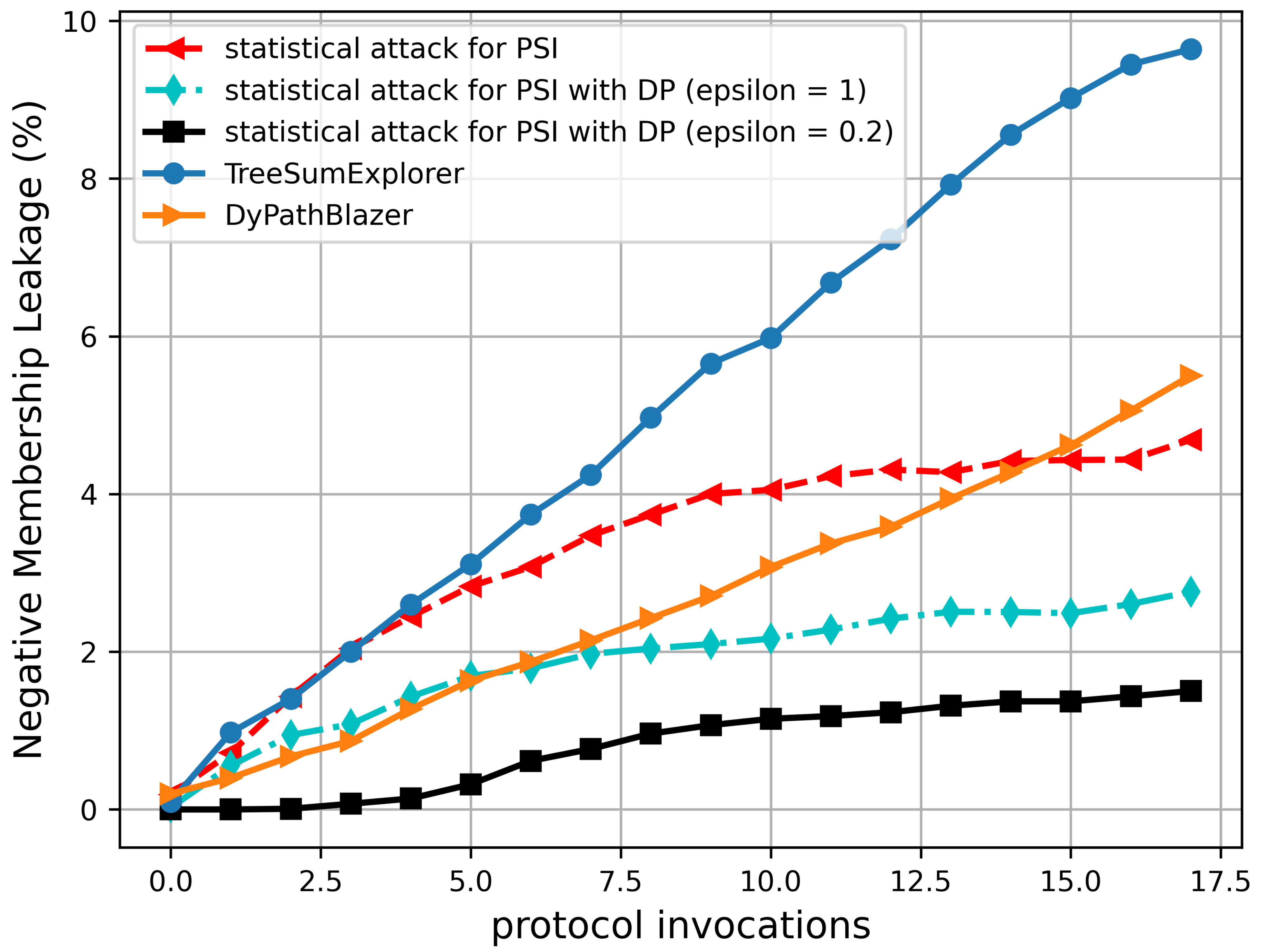
Statistic Approach - Bayesian Active Learning (ActBayesian)

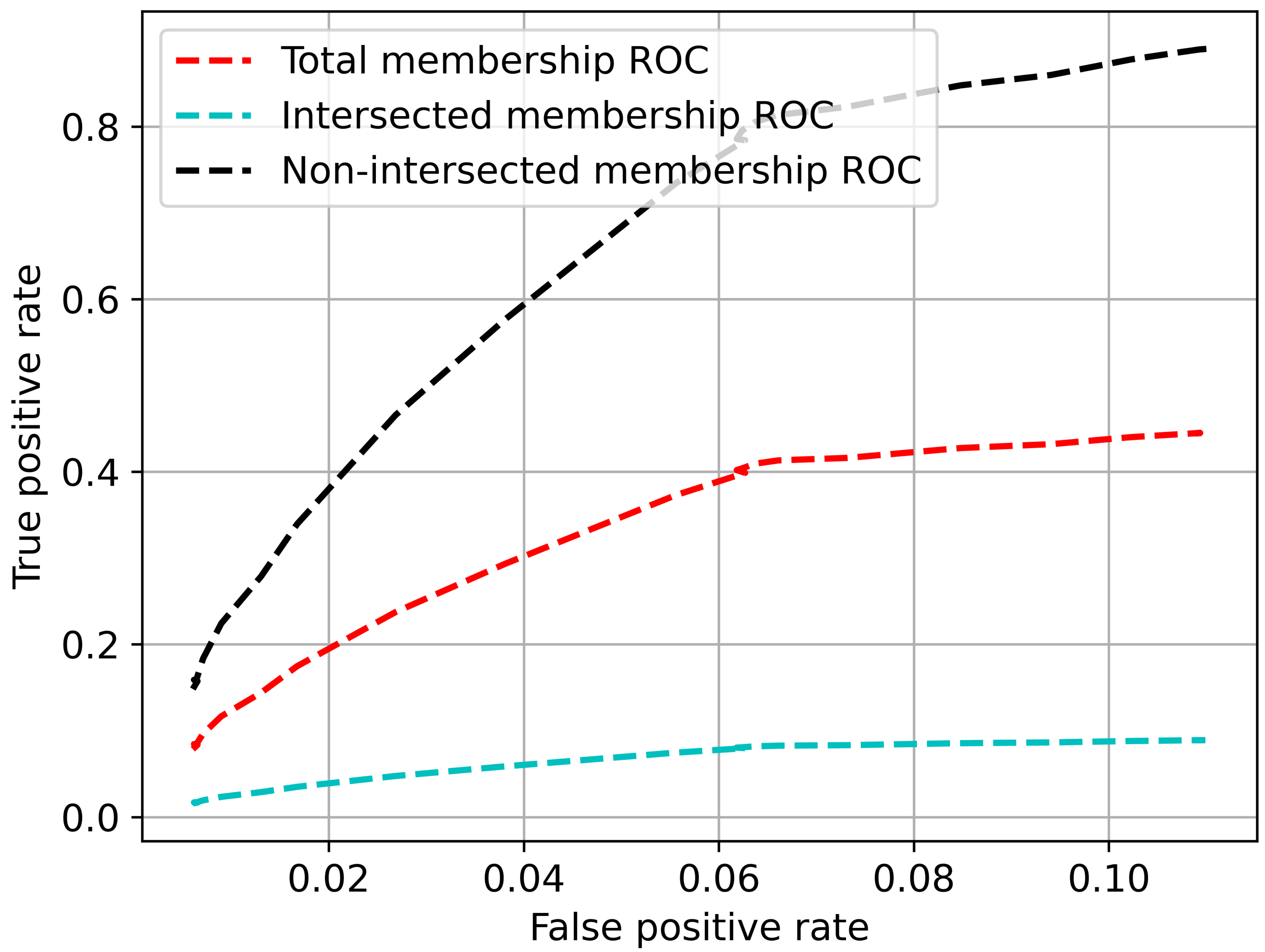
- Treat each element's membership as binary random variable
- Select inputs according to their distances to upper / lower threshold (with random sampling)
- Belief is updated using Bayesian posterior update



Experiments for PSI-CA with Differential Privacy (DP) protection

- Product company targets the advertising company
- Statistical attack remains valid even with DP protection





Experiments for PSI-CA with DP protection

	Default	Upper Threshold θ_u ↑ 0.8	1	Lower Threshold θ_l ↓ 0	0.2	Tolerance Factor tol ↑ 0	0.2	Sampling Rate r ↓ 0.3	0.9	Total Budget τ ↑ 10	50
True Positive Percentage	0.08	0.14	0.06 ↓	0.11 ↑	0.02	0.05	0.07 ↓	0.05 ↓	0.04	0.02	0.17 ↑
True Negative Percentage	0.27	0.25	0.29 ↑	0.22 ↓	0.31	0.23	0.25 ↓	0.21 ↓	0.22	0.12	0.83 ↑
Type I error rate	0.083	0.15	0 ↓	0.09 ↑	0.04	0.072	0.080 ↓	0.084 ↑	0.08	0.12	0.064 ↓
Type II error rate	0.085	0.087	0.092 ↑	0 ↓	0.17	0.077	0.082 ↓	0.085 —	0.082	0.16	0.055 ↓

Better Performance

- Error analysis under different parameters

Takeaways & Future works

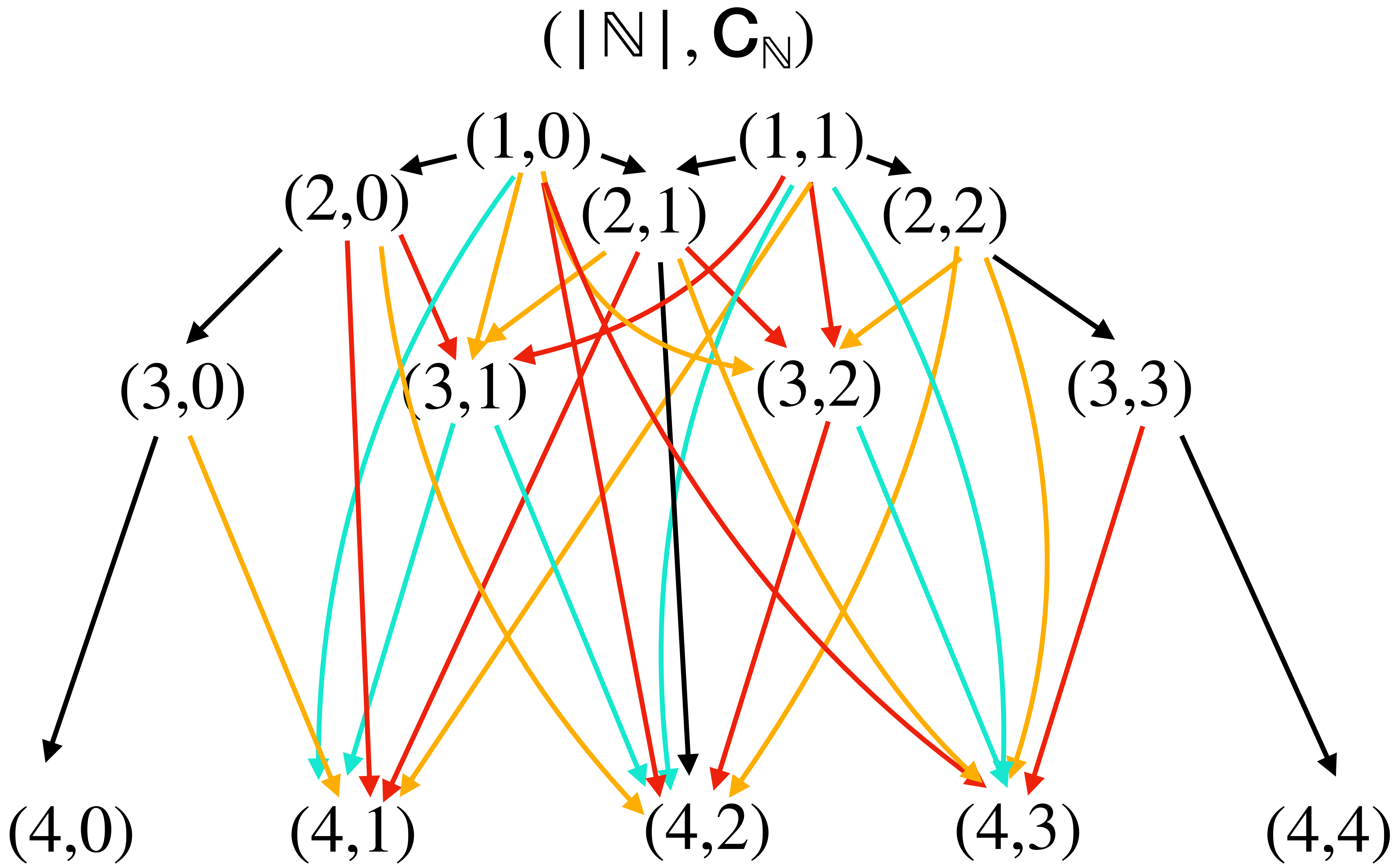
- Takeaways:
  - Most traditional MPC protocols are not sufficient to guarantee input privacy. Extra validations and privacy enhancements must be incorporated under stringent privacy regulation requirements.
  - Efficient attacks are able to make membership inference via a small number of protocol invocations (3% of the users in a datasets are re-identified within 5 PSI-SUM calls).
  - Leakage from statistical attack provides guidance for parameter determination for counter measures, such as the privacy budget for DP.
- Future works:
  - Rethinking security model of PSI protocol to leverage membership leakage.
  - Defenses against proposed attacks: ML to detect pattern, DP in high privacy regime, etc.

The End

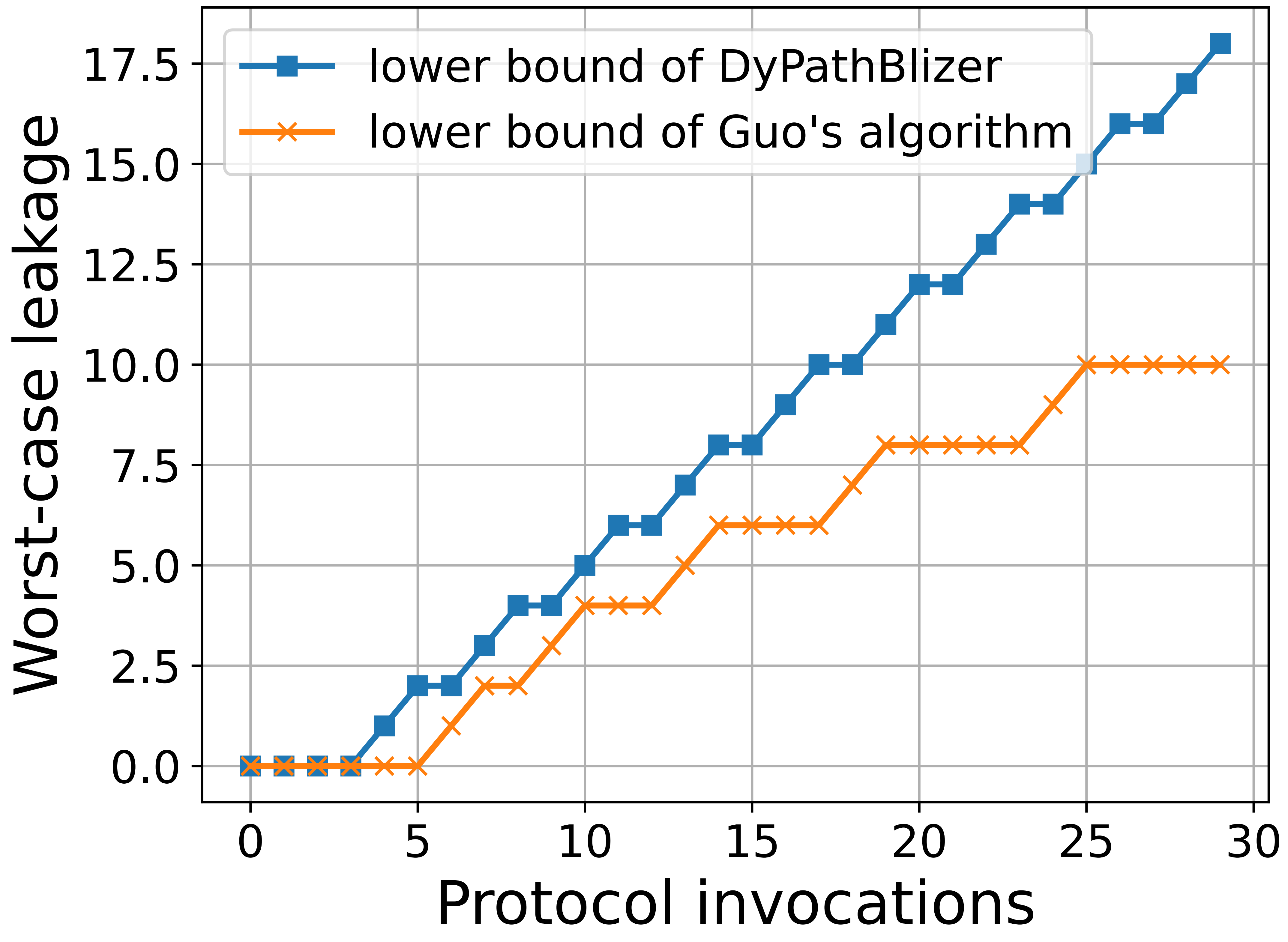
Q/A

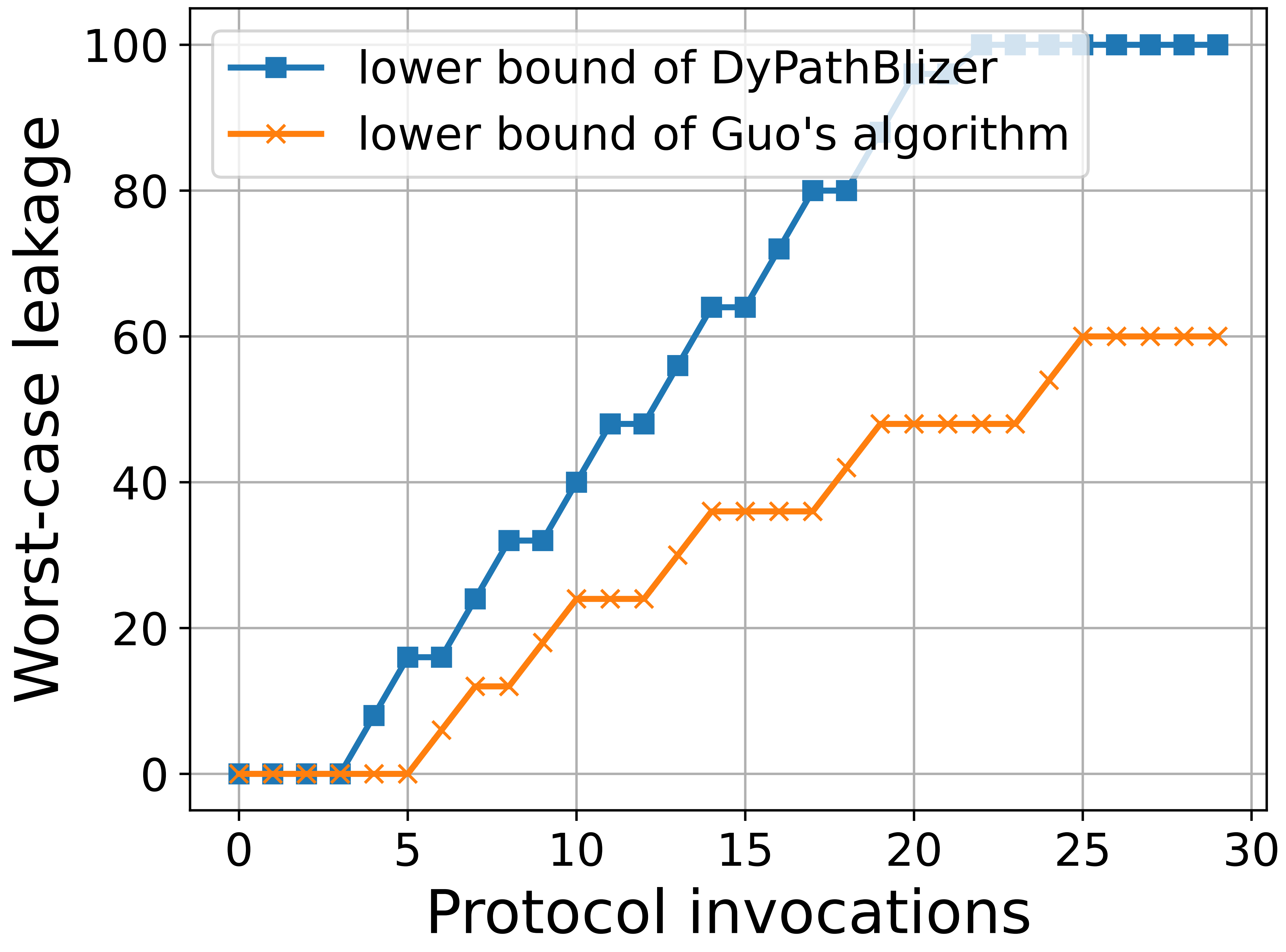
Backup slides - Illustration of backtracking

- $\Theta(|\mathbb{N}|, C_{\mathbb{N}}, \tau)$ also memorizes maximum expected leakage
- $\Gamma(|\mathbb{N}|, C_{\mathbb{N}})$ denotes the expected protocol call needed to infer $|\mathbb{N}|$ individuals. Γ is derived using dynamic programming.



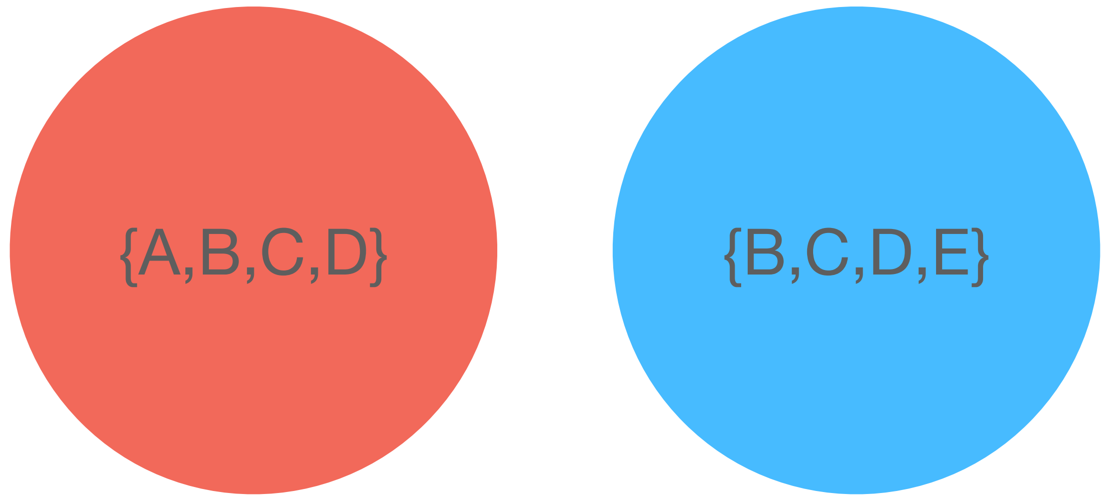
Backup slides - Lower bound comparison, DyPathBlazer v.s. Baseline

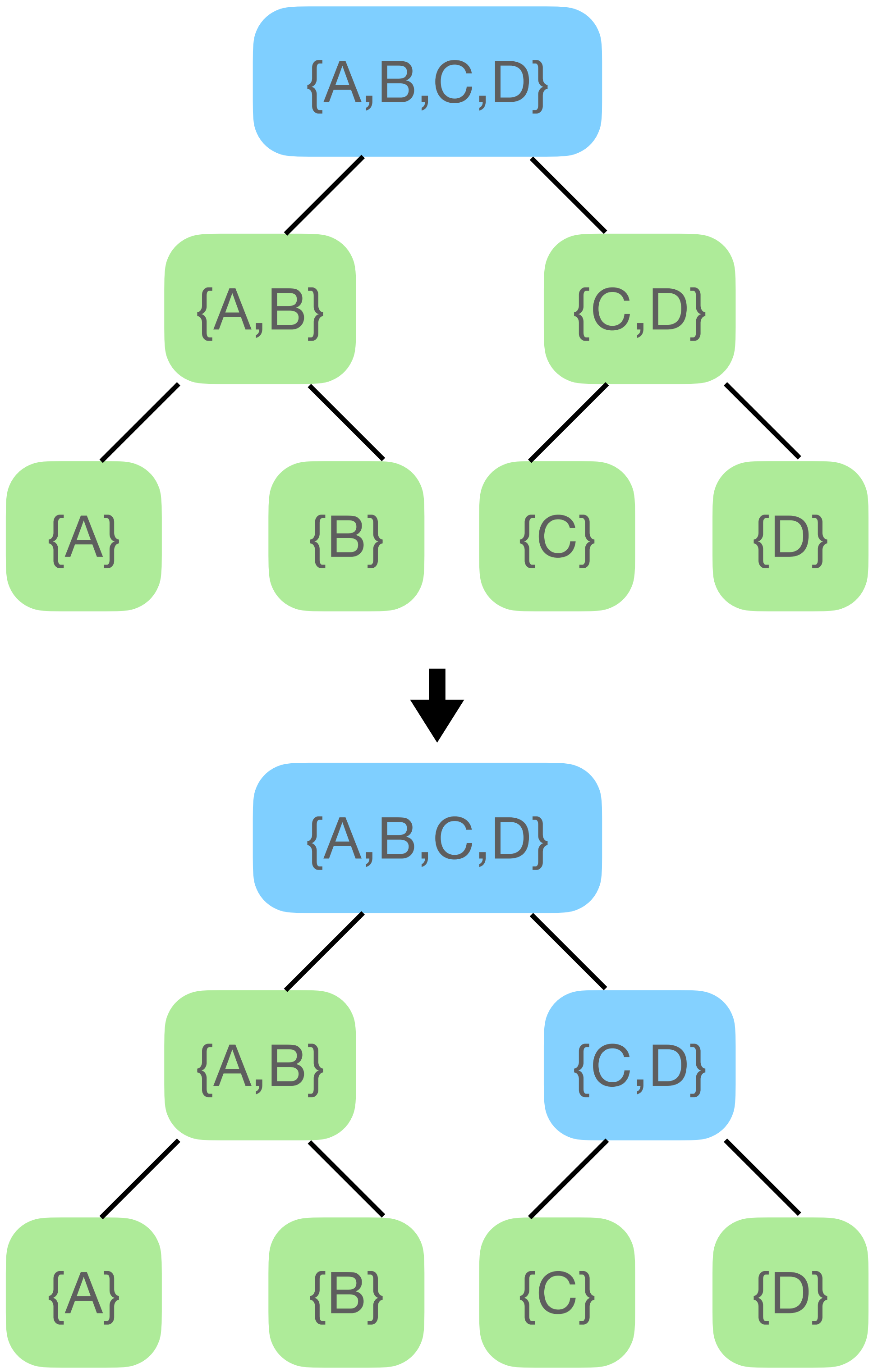




Both cases consider a dataset of 100 individuals. Case 1) assumes 50 positive members, case 2) assumes 10 positive members. Higher lower bounds from DyPathBlazer guarantees better efficiency in the worst-case scenario.

Backup slides - Steps in the baseline algorithm





$m = 3 \quad Z = \varnothing$ T= {A,B,C,D}

Priority Queue: (3/4, (3, {A,B,C,D}))

pop()

$m_{node} = 3$ node = {A,B,C,D} $|T_{node}| = 4$ L= {A,B}, R={C,D}

$m_L = 1 \quad m_R = 2$

push

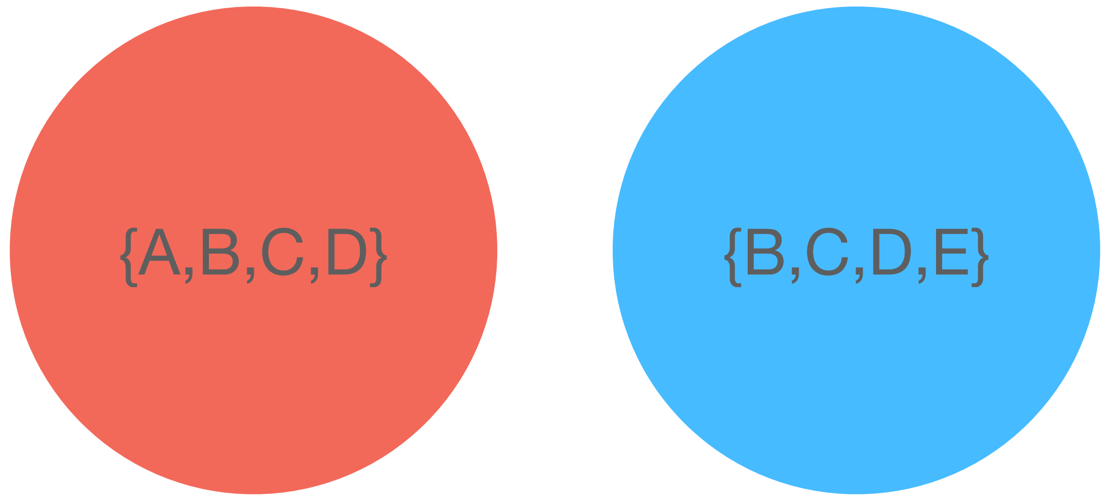
Priority Queue: (1/2, (1, {A,B}))

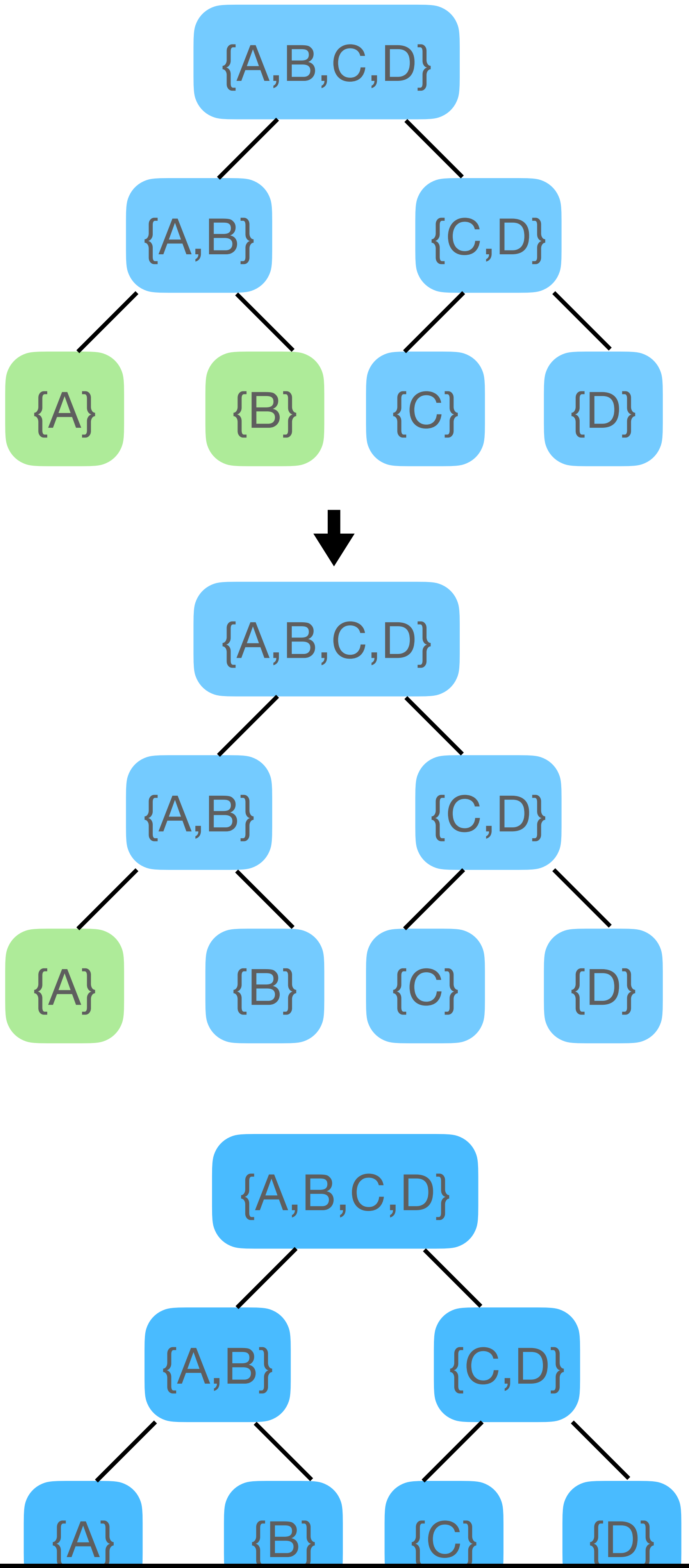
$m_{node} = 2$ node = {C,D} $|T_{node}| = 2$ L= {C}, R={D}

Stops because $m_{node} = |T_{node}|$

$Z = \{C, D\}$

Backup slides - Steps in the baseline algorithm (cont'd)





Priority Queue: (1/2, (1, {A,B}))

↓ pop()

$m_{node} = 1$ node = {A,B} $|T_{node}| = 2$ L= {A}, R={B}

↓

$m_L = 0$ $m_R = 1$

↓ push

Priority Queue: (0, (0, {A}))

↓

$m_{node} = 1$ node = {B} $|T_{node}| = 1$ L= {}, R={}

↓

Stops because $m_{node} = |T_{node}|$

$Z = \{B, C, D\}$

Priority Queue: (0, (0, {A}))

↓ pop()

$m_{node} = 0$ node = {A} $|T_{node}| = 1$ L= {0}, R={0}

↓

Stops because $m_{node} = 0$