Deletions and Dishonesty

PDS in Adversarial Settings

Mia Filić

Joint work with Keran Kocher Ella Kummer Anupama Unnikrishnan

ETH Zurich

Probabilistic Data Structures (PDS)

A way to

compactly represent (tons of) data

and

provide approximate **answers to queries** about that data

Probabilistic Data Structures (PDS)

A way to

compactly represent (tons of) data

and

provide approximate **answers to queries** about that data

- Frequency estimation
  How many times does x appear in the data?
  Count-min sketch, HeavyKeeper

Probabilistic Data Structures (PDS)

A way to

compactly represent (tons of) data

and

provide approximate **answers to queries** about that data

- Frequency estimation
  How many times does x appear in the data?
  Count-min sketch, HeavyKeeper

- Membership queries
  Is x in the set?
  Bloom filter, Cuckoo filter, Counting filter

Probabilistic Data Structures (PDS)

A way to

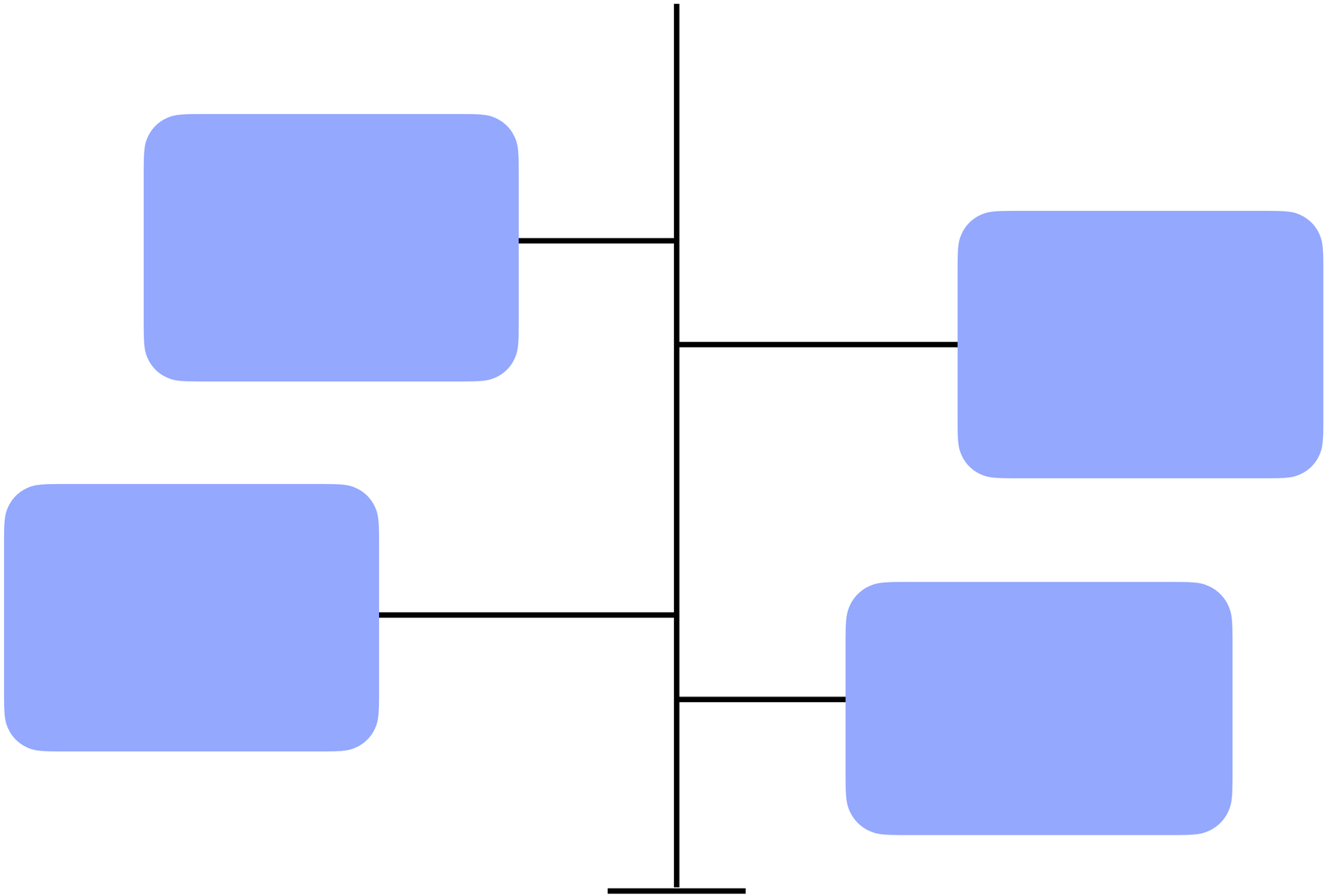
compactly represent (tons of) data

and

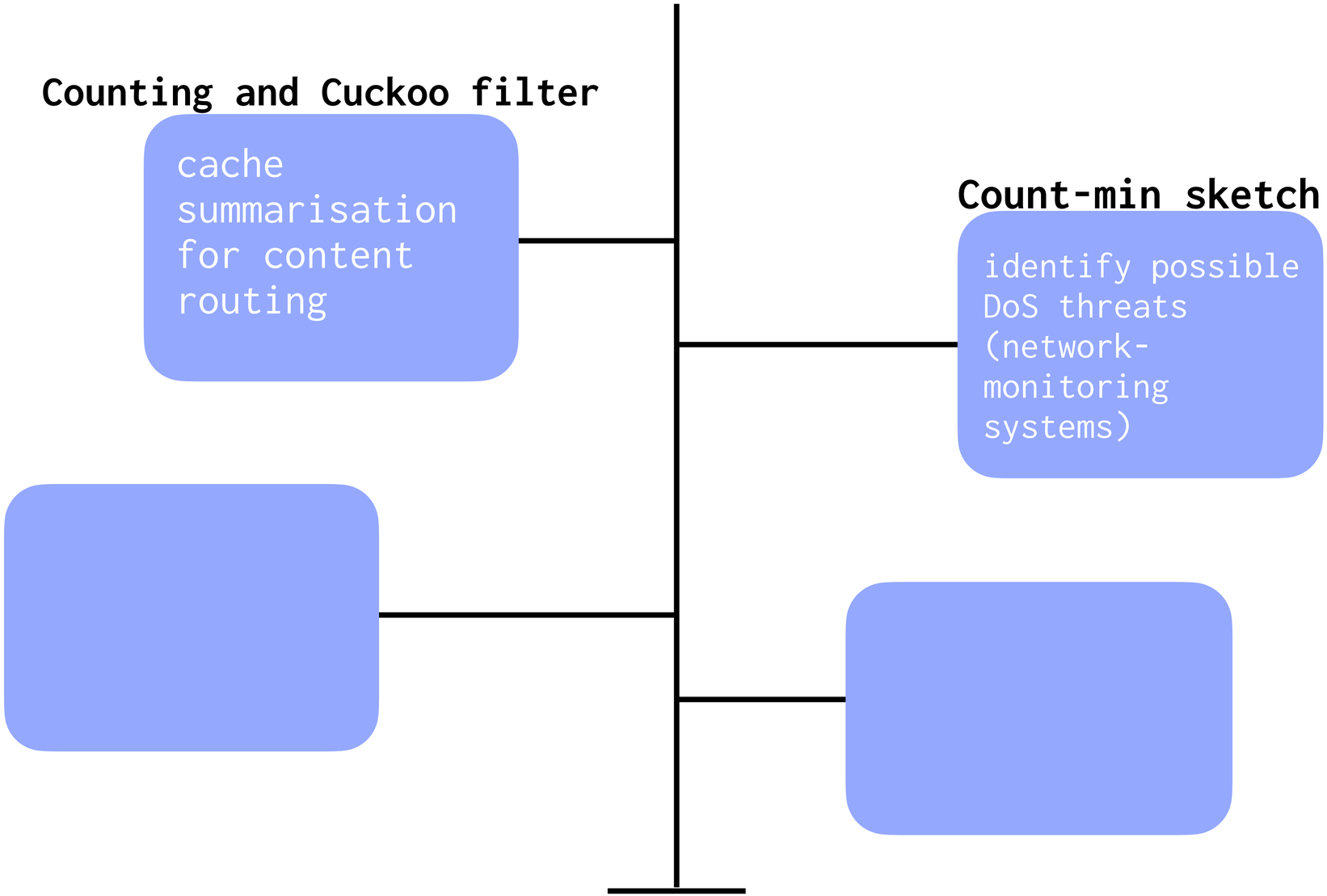
provide approximate **answers to queries** about that data

- Frequency estimation
  How many times does x appear in the data?
  Count-min sketch, HeavyKeeper
- Membership queries
  Is x in the set?
  Bloom filter, Cuckoo filter, Counting filter
- Cardinality estimation
  How many distinct elements are in the set?
  HyperLogLog, KMV estimator

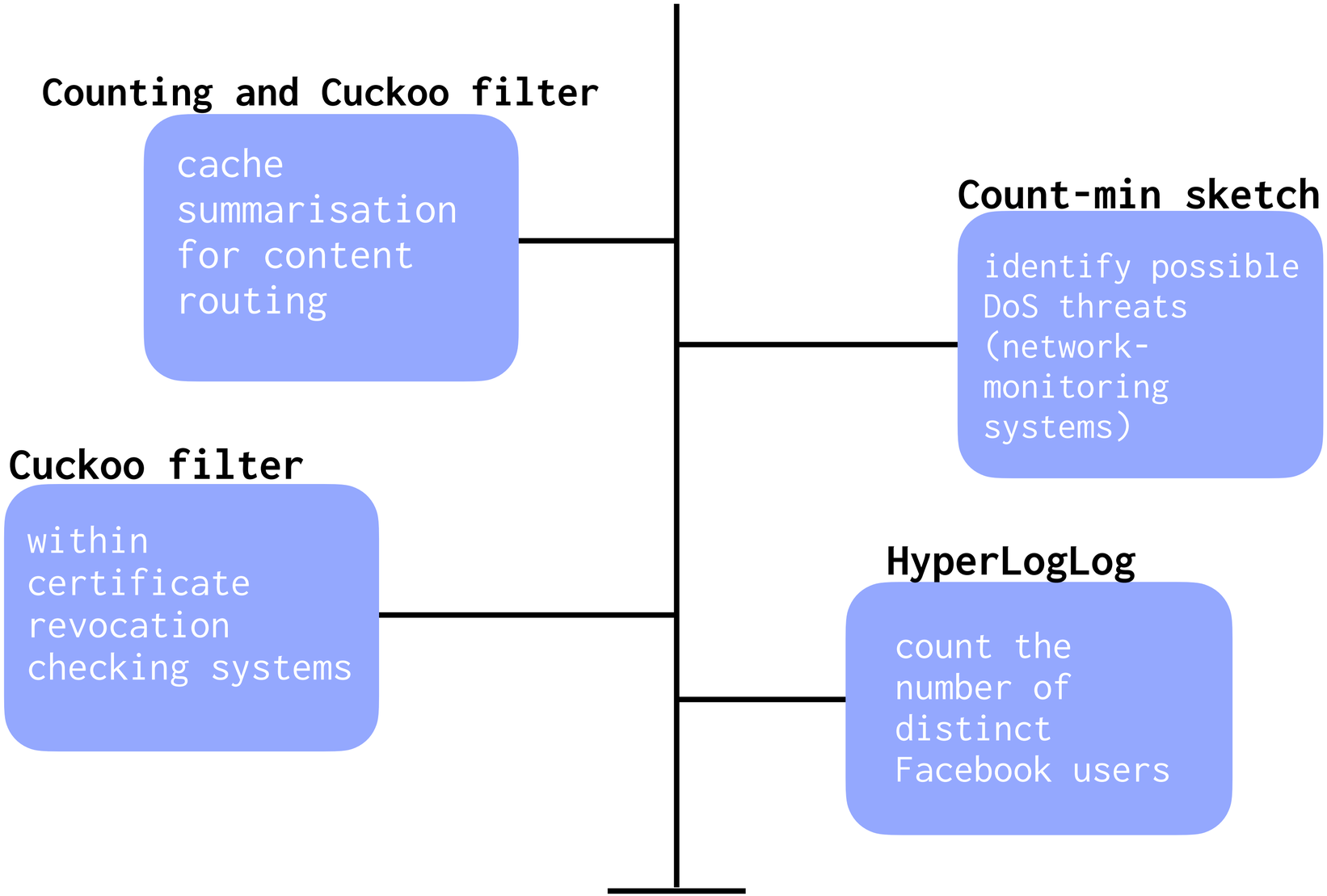
Where are PDS used?



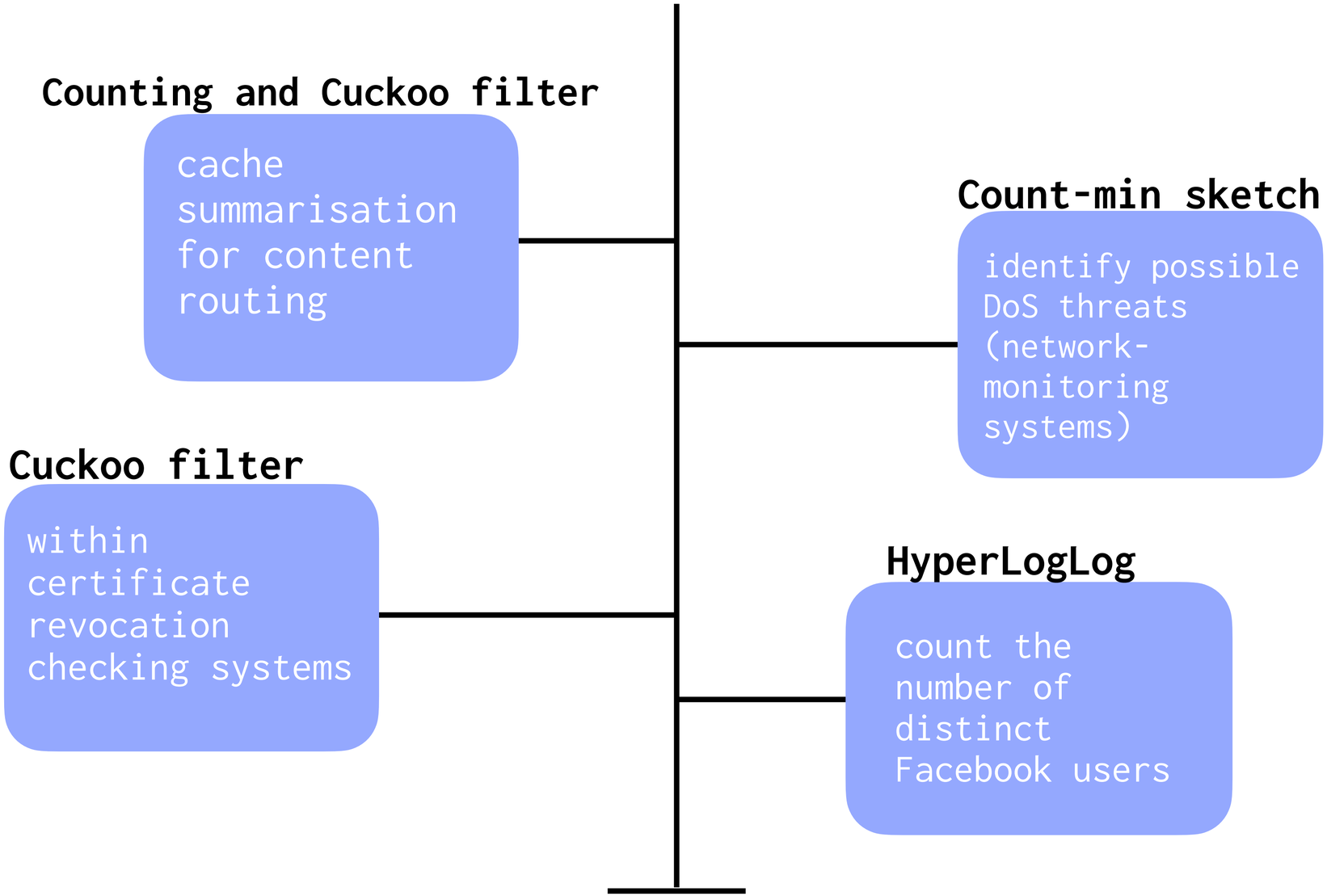
Where are PDS used?



Where are PDS used?



What can go wrong?



PDS in adversarial settings

Adversarial correctness

- Adversary can **interfere** with the correct functionality of the PDS

PDS in adversarial settings

Adversarial correctness

- Adversary can **interfere** with the correct functionality of the PDS

Privacy

- Adversary could try to **learn** about the elements represented by the PDS

PDS in adversarial settings

Adversarial correctness

Privacy

- Adversary can **interfere** with the correct functionality of the PDS
- Adversary could try to **learn** about the elements represented by the PDS (e.g., [FPUV22])

PDS in adversarial settings

Adversarial correctness

Privacy

Secure PDS

- Adversary can **interfere** with the correct functionality of the PDS
- Adversary could try to **learn** about the elements represented by the PDS (e.g., [FPUV22])
- How can we **provably protect** PDS in adversarial settings?

This work

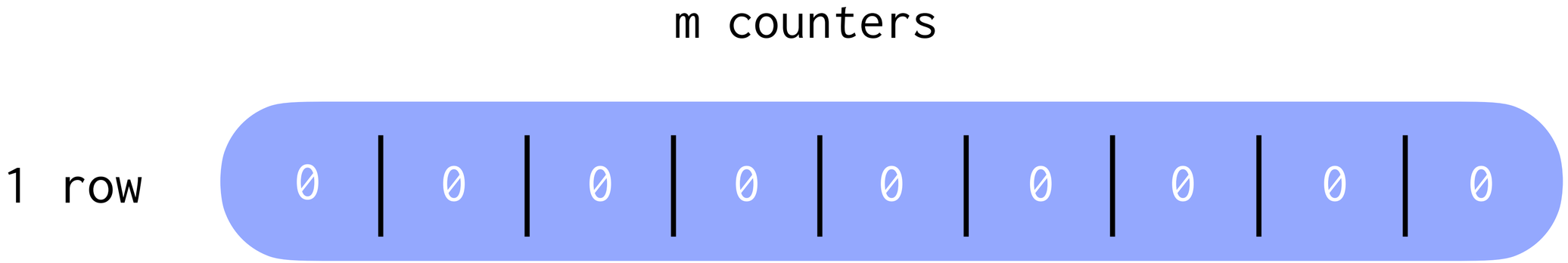
Adversarial correctness

Privacy

Secure AMQ-PDS W/ deletions

- **Adversary can interfere with the correct functionality of a class of AMQ-PDS w/ deletions, e.g.,Counting and Cuckoo Filters**
- Adversary could try to **learn** about the elements represented by the PDS (e.g.,[FPUV22])
- **How can we provably protect e.g. Counting and Cuckoo Filters in adversarial settings?**

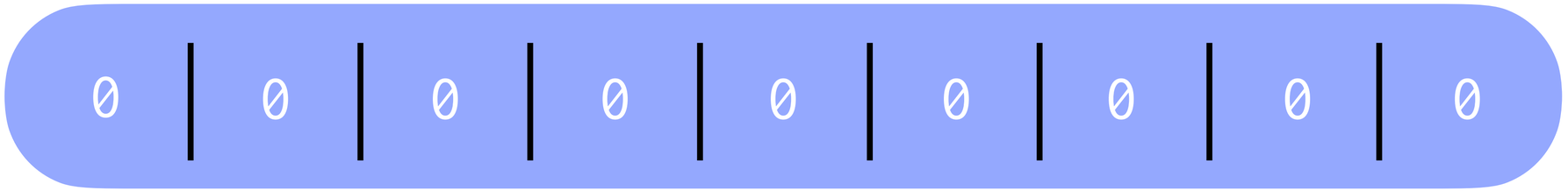
Counting filter



Counting filter

m counters

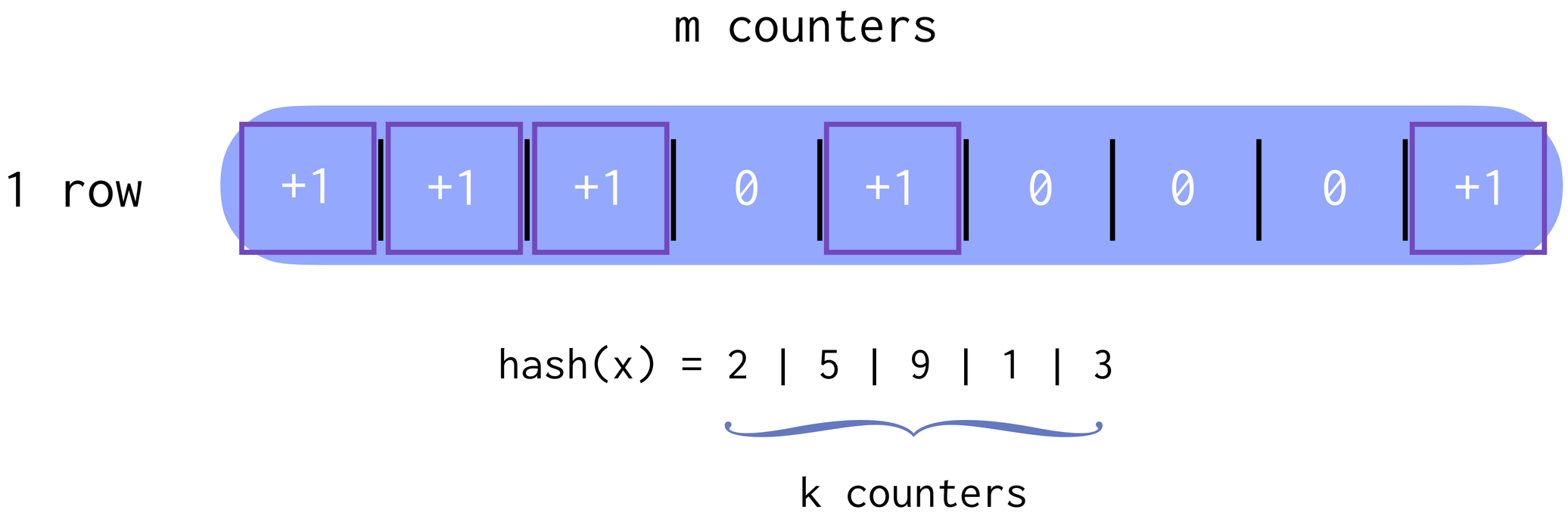
1 row



hash(x) = 2 | 5 | 9 | 1 | 3

k counters

Counting filter: insert(x)



Counting filter: insert(..)

m counters

1 row

hash(x) = 2 | 5 | 9 | 1 | 3

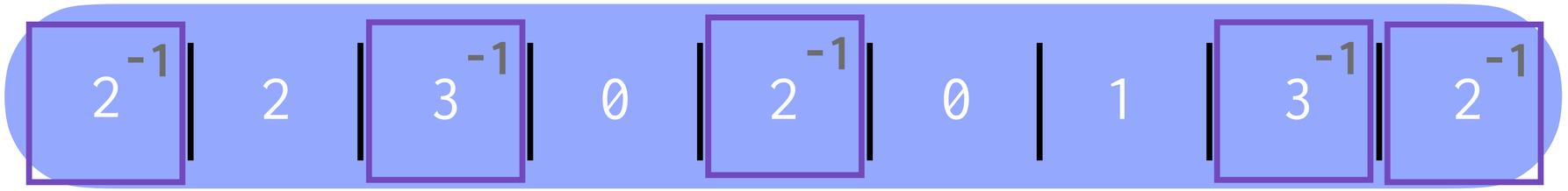
hash(y) = 8 | 9 | 3 | 1 | 5

hash(z) = 7 | 8 | 3 | 2 | 8

Counting filter: delete(y)

m counters

1 row



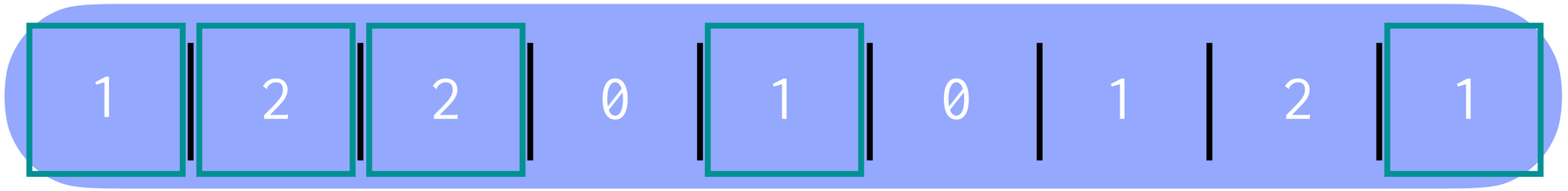
hash(y) = 8 | 9 | 3 | 1 | 5

k counters

Counting filter: query(x)

m counters

1 row



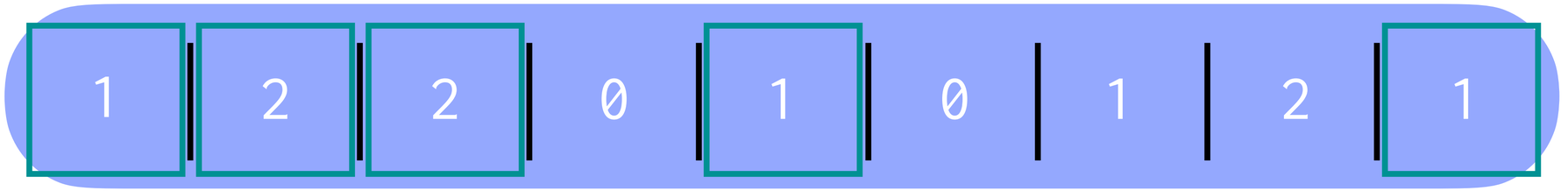
hash(x) = 2 | 5 | 9 | 1 | 3

k counters

Counting filter: query(x)

m counters

1 row



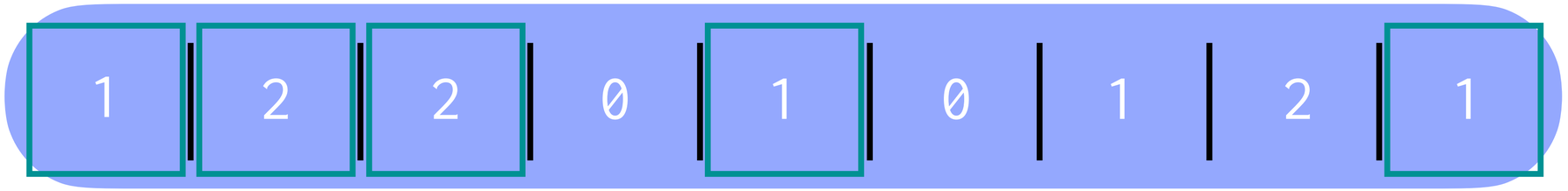
hash(x) = 2 | 5 | 9 | 1 | 3

CF(x) = [all cnt(x) > 0]

Counting filter: query(x)

m counters

1 row



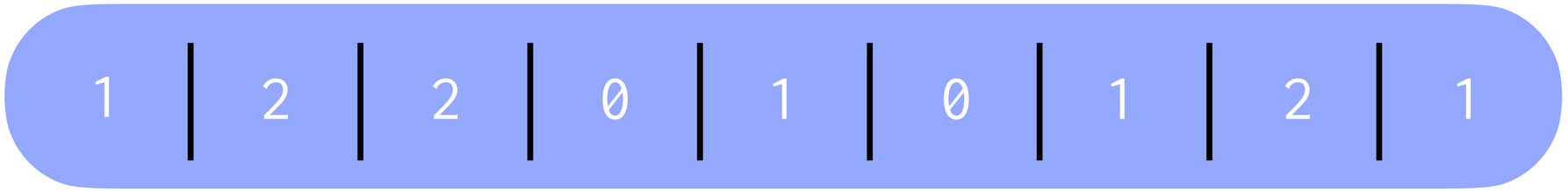
hash(x) = 2 | 5 | 9 | 1 | 3

CF(x) = T

Counting filter: query(x)

m counters

1 row

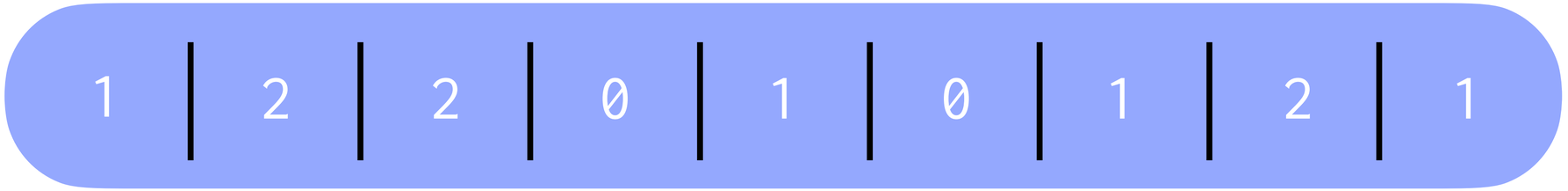


False positives and negatives ?

Counting filter: query(x)

m counters

1 row



size of filter

number of elements in filter

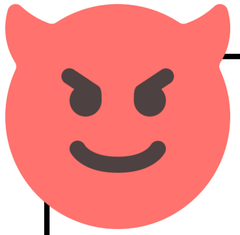
False positives: Pr[FP] = f(m, k, n)

number of hash functions

Honest setting only!

False negatives: often assumed not to occur

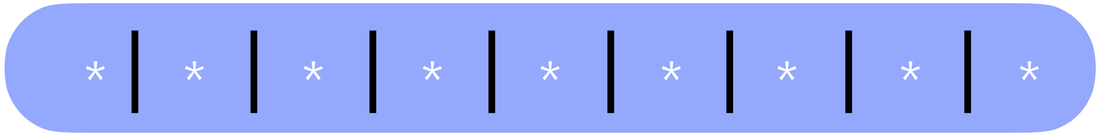
What can go wrong in adversarial settings?



CF[H]:
insert
query
delete

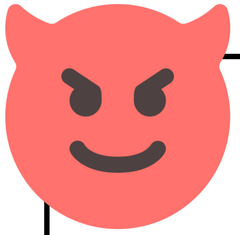
hash

User



hash(.)

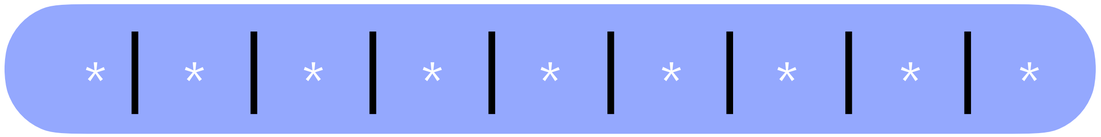
What can go wrong in adversarial settings?



CF[H]:
insert
query
delete

hash

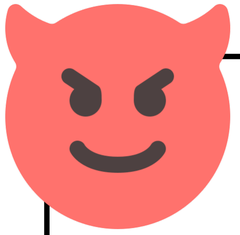
User



hash(.)

Public hash functions = precomputation attacks

What can go wrong? [CPS19]

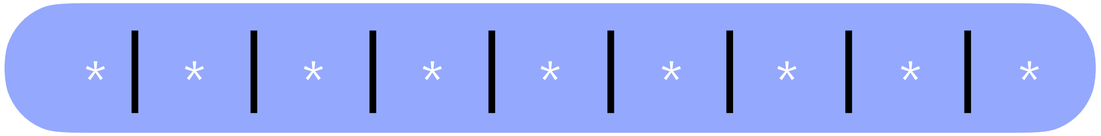


CF[H]:
insert
query
delete

hash

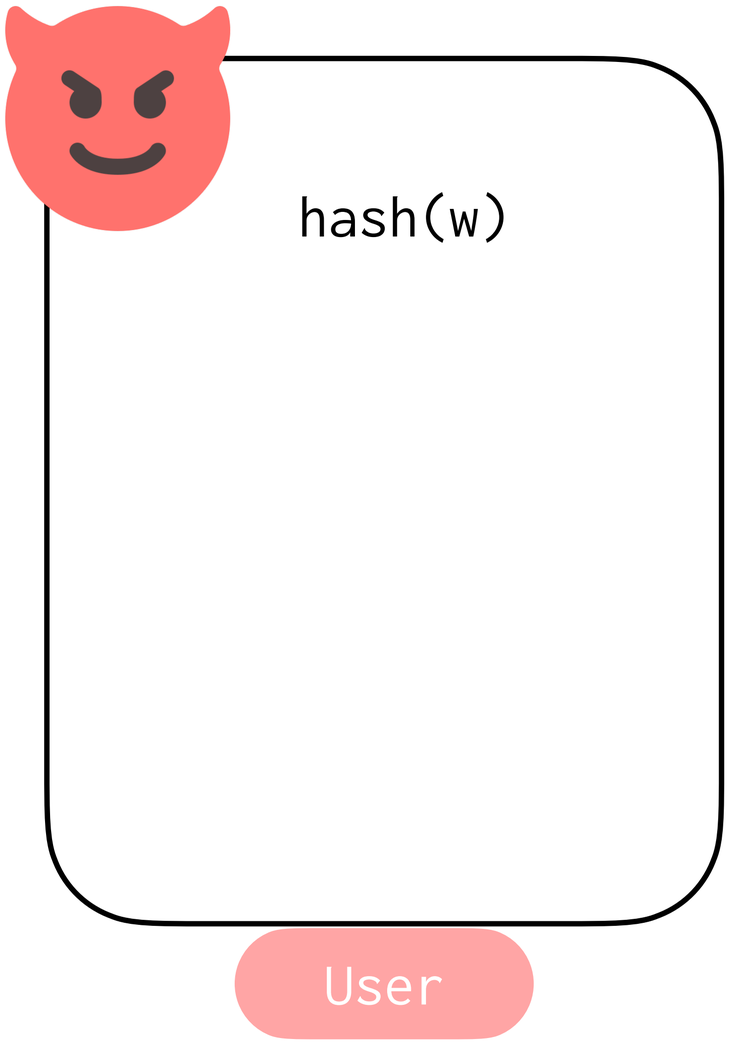
goal: make target element w a false positive

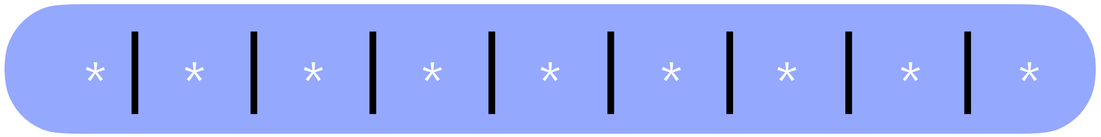
User



hash(.)

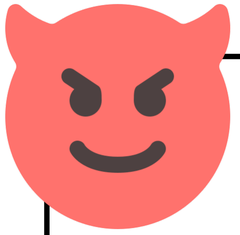
What can go wrong?





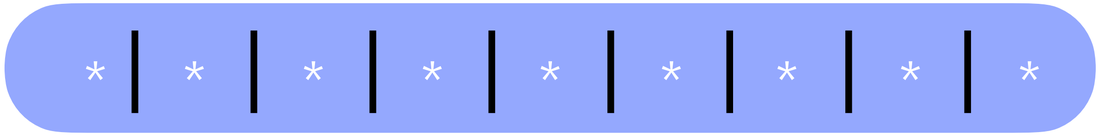
hash(w) = 1 | 3 | 9 | 2 | 8

What can go wrong?



hash(w)
hash(x)
hash(y)
hash(z)

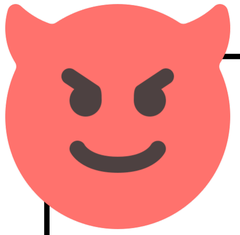
User



hash(w) = **1** | **3** | 9 | 2 | 8

hash(x) = 2 | 5 | 9 | **1** | **3**
hash(y) = 8 | 9 | **3** | **1** | 5

What can go wrong?



hash(w)
hash(x)
hash(y)
hash(z)

insert x
insert y

query(w) = T

User

> 0	> 0	> 0	*	*	*	*	> 0	> 0

hash(w) = **1** | **3** | 9 | 2 | 8

hash(x) = 2 | 5 | 9 | **1** | **3**
hash(y) = 8 | 9 | **3** | **1** | 5

What could go wrong, beyond the paper focus?

Bloom filter

tamper with the false positive probability

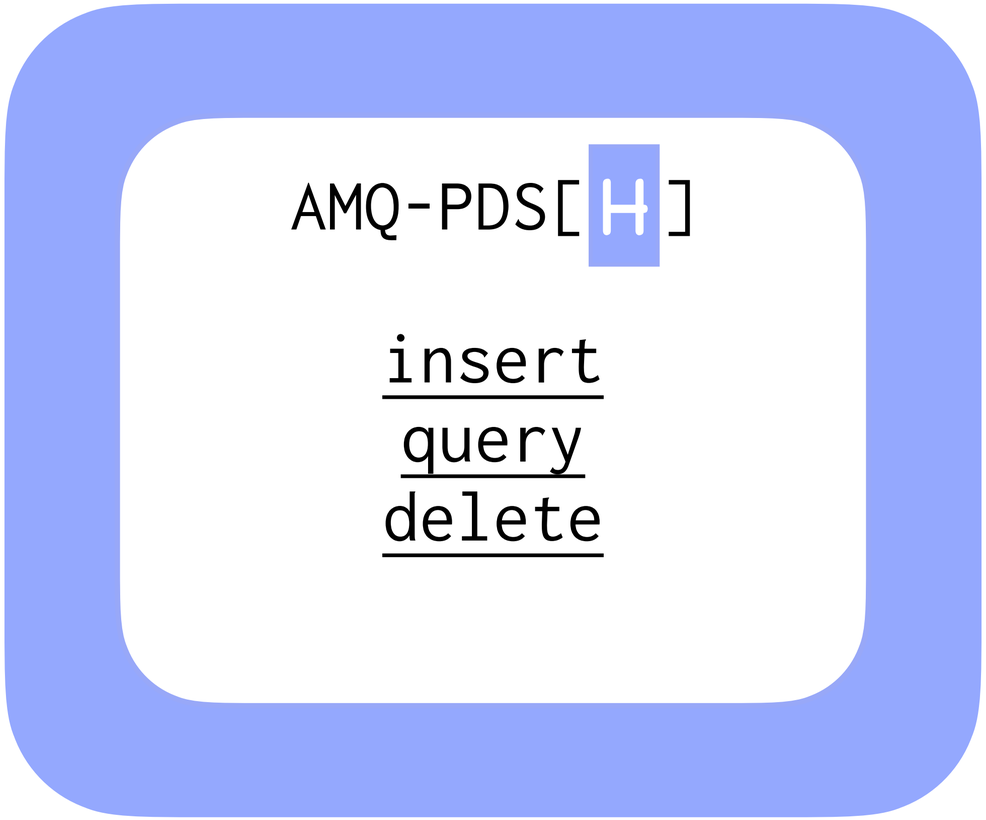
HyperLogLog

inflate the cardinality estimate [PR22]

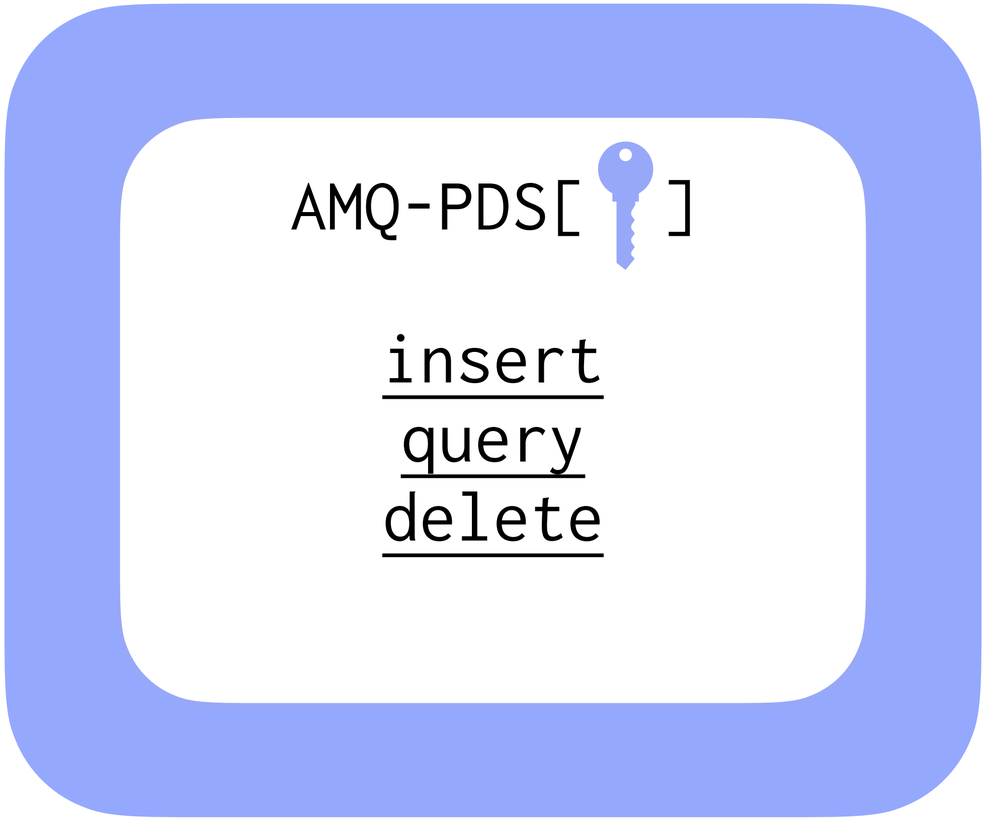
Count-min sketch

cause frequency of an element to be unexpectedly overestimated [MFS23]

How do we define secure PDS?



How do we define secure PDS?



How do we define secure PDS?

Game-based notions
[NY15, CPS19]

Specific adversarial goal

AMQ-PDS[]

insert
query
delete

How do we define secure PDS?

Game-based notions
[NY15, CPS19]

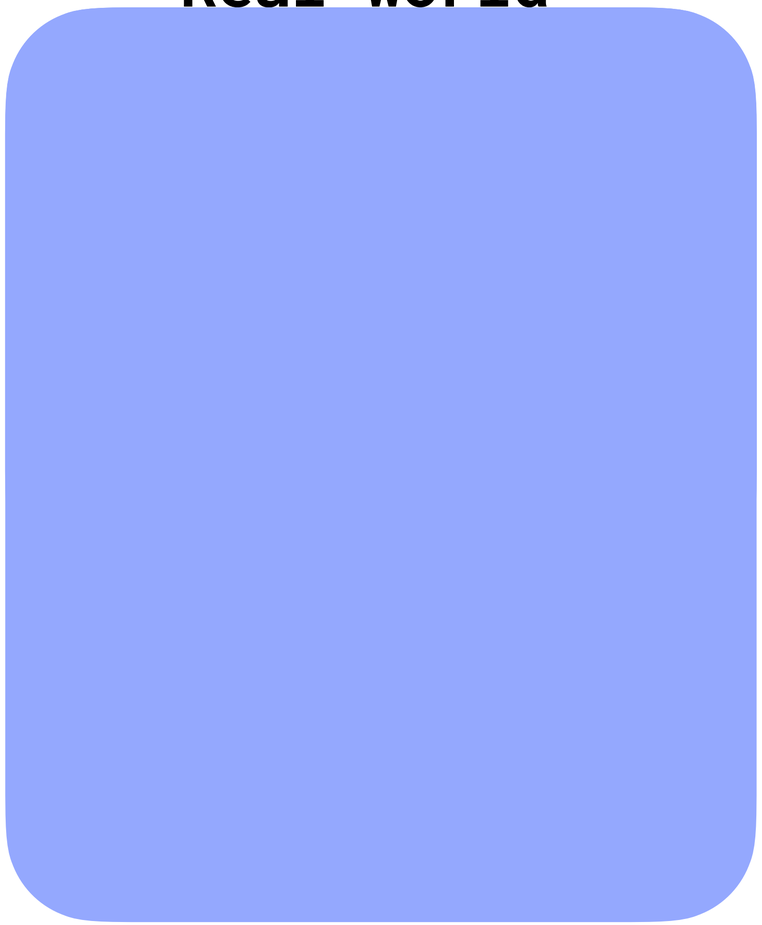
Specific adversarial goal

Simulation-based notions
[PR22, FPUV22]

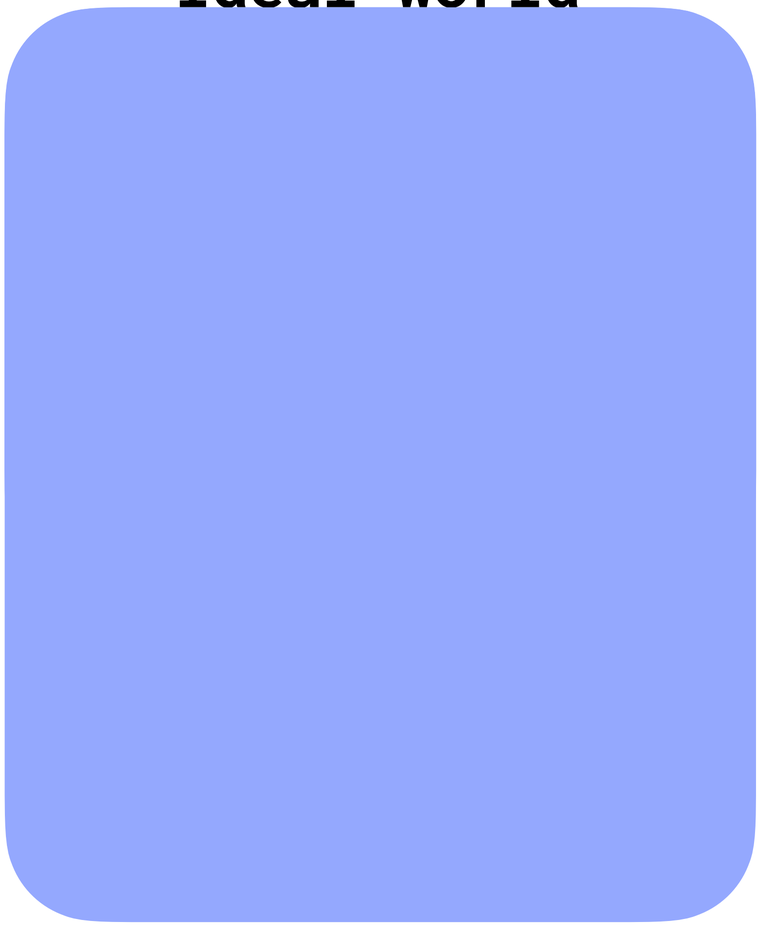
Any adversarial goal

Simulation-based framework

Real world



Ideal world



Simulation-based framework

Real world

AMQ-PDS[]

insert
query
delete

adversary interacts with a concrete AMQ-PDS

Ideal world

Simulator

insertSim
querySim
deleteSim

adversary interacts with a simulator

Simulation-based framework

we want to analyse this ->

Real world

AMQ-PDS[]

insert
query
delete

adversary interacts with a concrete AMQ-PDS

Ideal world

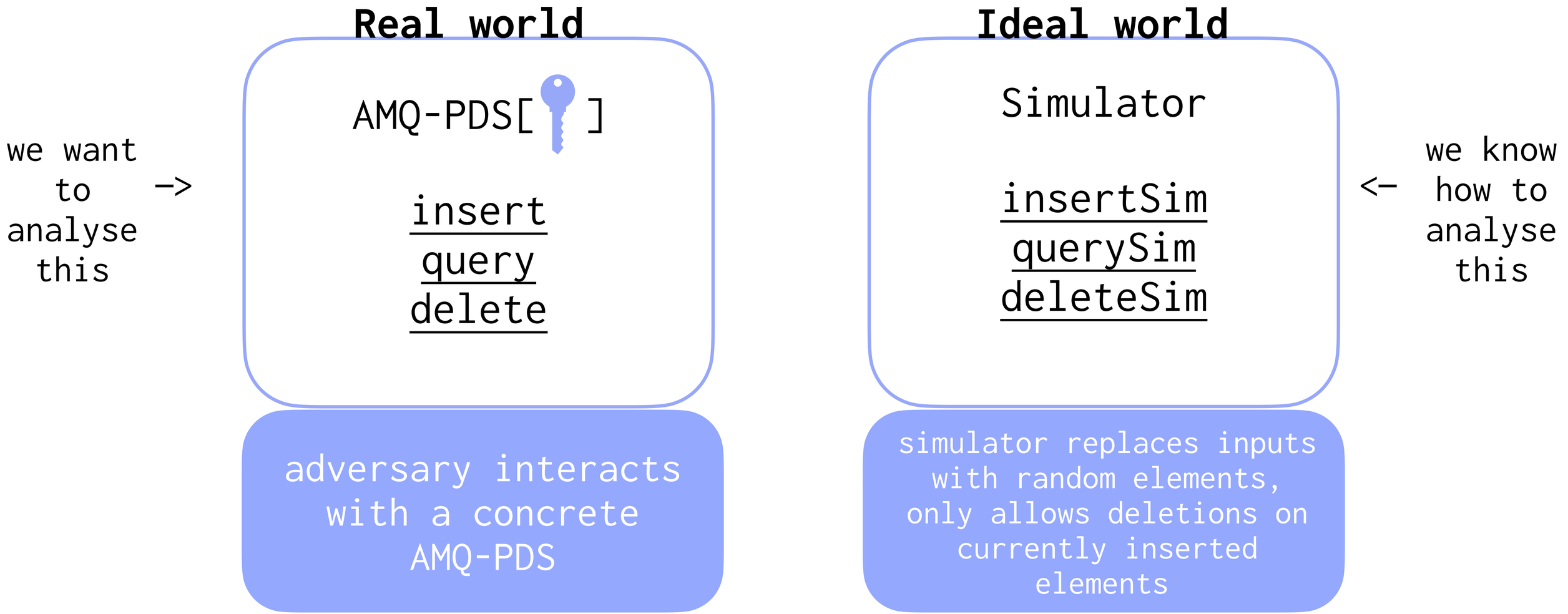
Simulator

insertSim
querySim
deleteSim

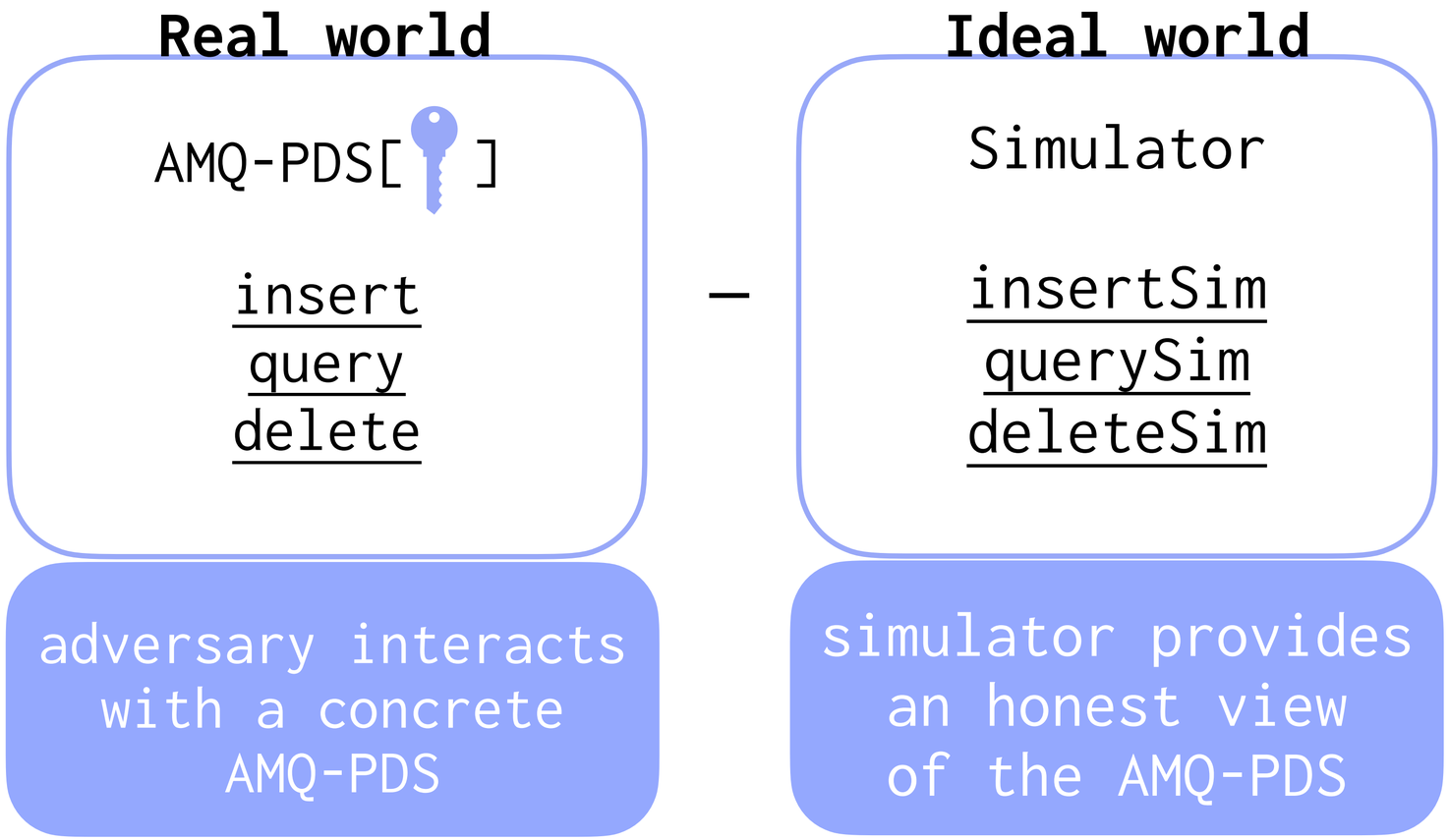
simulator provides an honest view of the AMQ-PDS

<- we know how to analyse this

Adversarial correctness



Adversarial correctness



Adversarial correctness

Real world

AMQ-PDS[]

insert
query
delete

adversary interacts with a concrete AMQ-PDS

−

Ideal world

Simulator

insertSim
querySim
deleteSim

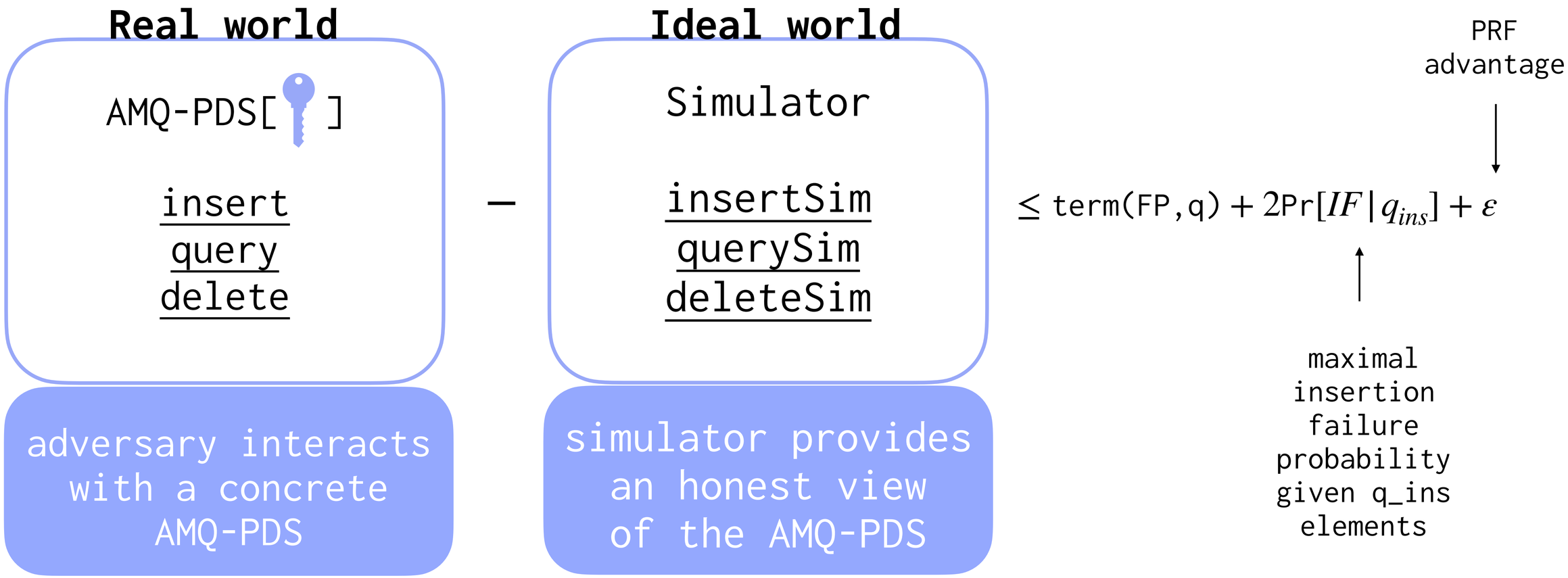
simulator provides an honest view of the AMQ-PDS

function of the number of different oracle calls

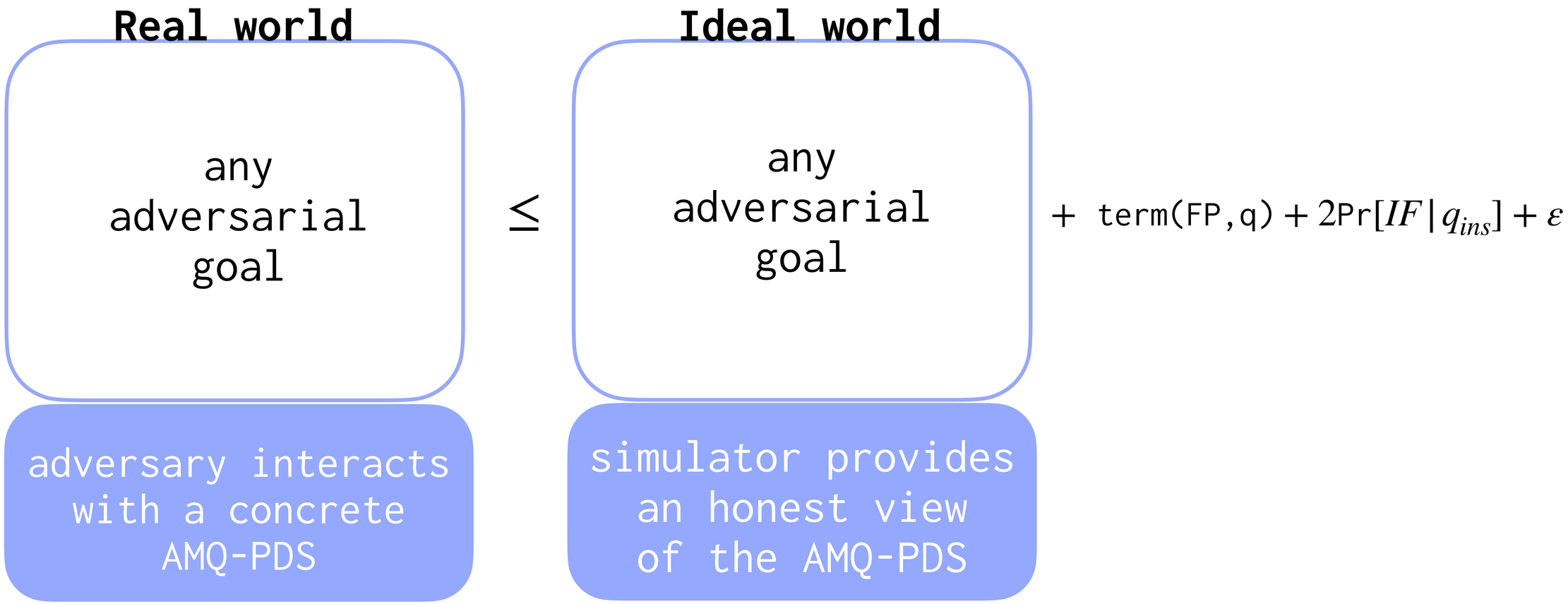
maximal false positive probability given q_ins elements

$$\leq \underbrace{(q_{ins} + 2q_{qry} + q_{del}) \cdot \Pr[FP \mid q_{ins}]}_{\text{term(FP,q)}} + \ldots$$

Adversarial correctness



Adversarial correctness



Adversarial correctness

Real world

finding a false positive given q_ins insertions, q_del deletions, q_qry queries

adversary interacts with a concrete AMQ-PDS

$\leq$

Ideal world

finding a false positive given q_ins insertions, q_del deletions, q_qry queries

simulator provides an honest view of the AMQ-PDS

$+ \ \mathrm{term}(\mathrm{FP}, q) + 2\Pr[IF \mid q_{ins}] + \varepsilon$

Adversarial correctness

Real world

finding a false positive given q_ins insertions, q_del deletions, q_qry queries

adversary interacts with a concrete AMQ-PDS

$$\leq \quad \Pr[FP \mid q_{ins}] \quad + \text{term}(\text{FP},\text{q}) + 2\Pr[IF \mid q_{ins}] + \varepsilon$$

Adversarial correctness

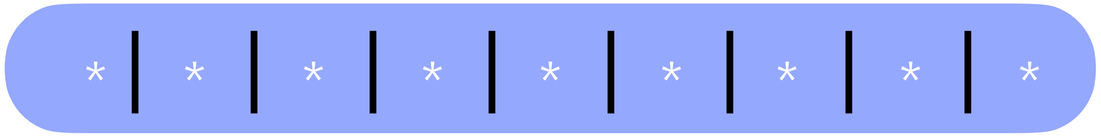
Real world

finding a false positive given q_ins insertions, q_del deletions, q_qry queries

adversary interacts with a concrete AMQ-PDS

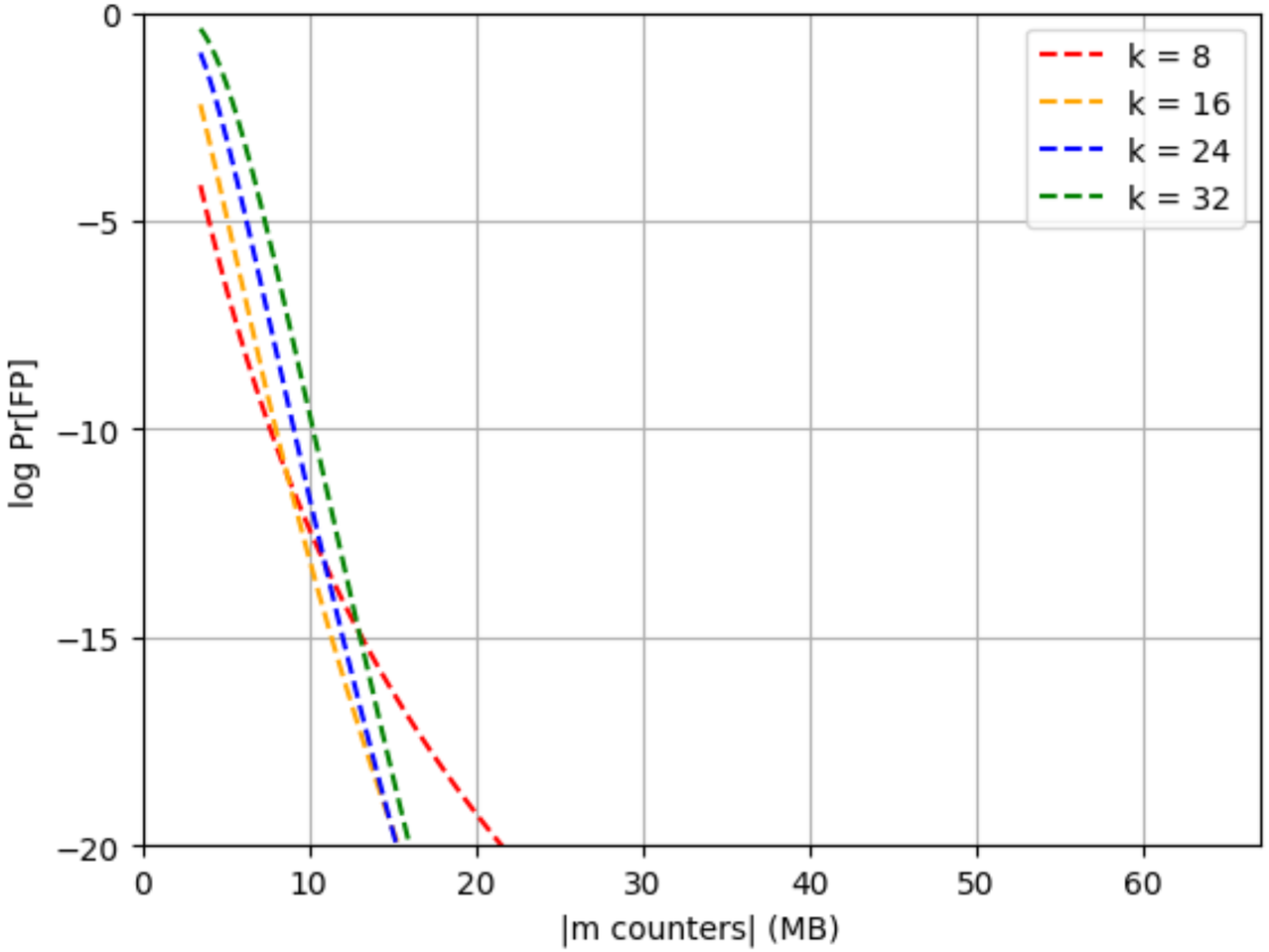
$\leq \quad \Pr[FP \mid q_{ins}] \quad + \text{term}(\text{FP},\text{q}) + 2\Pr[IF \mid q_{ins}] + \varepsilon$

Counting filter



PRF(.)

Securing Counting filters in practice

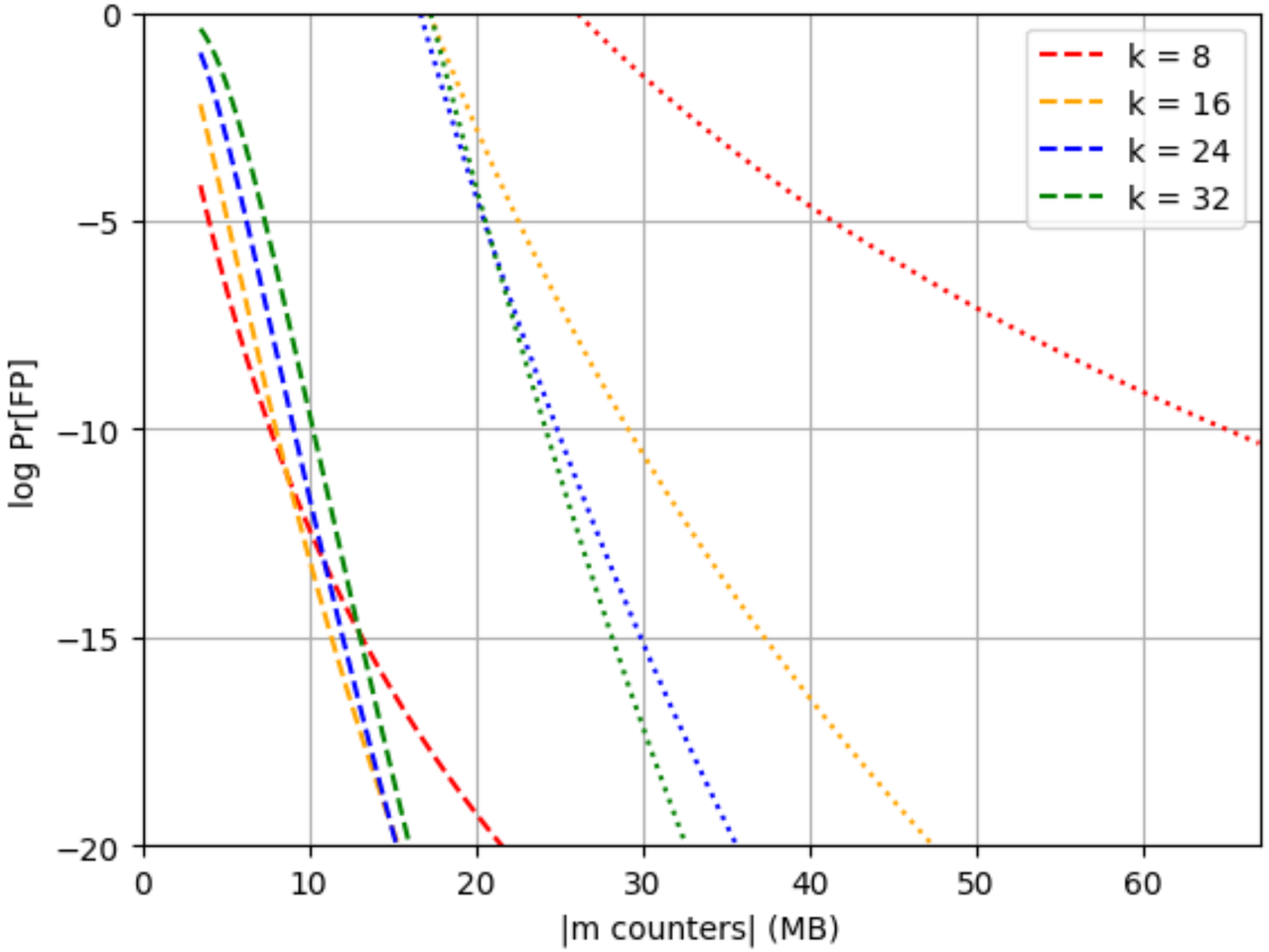


------- honest setting

Maximum counter value is 15

User makes 2^{20} insertions, deletions and queries

Securing Counting filters in practice



------- honest setting

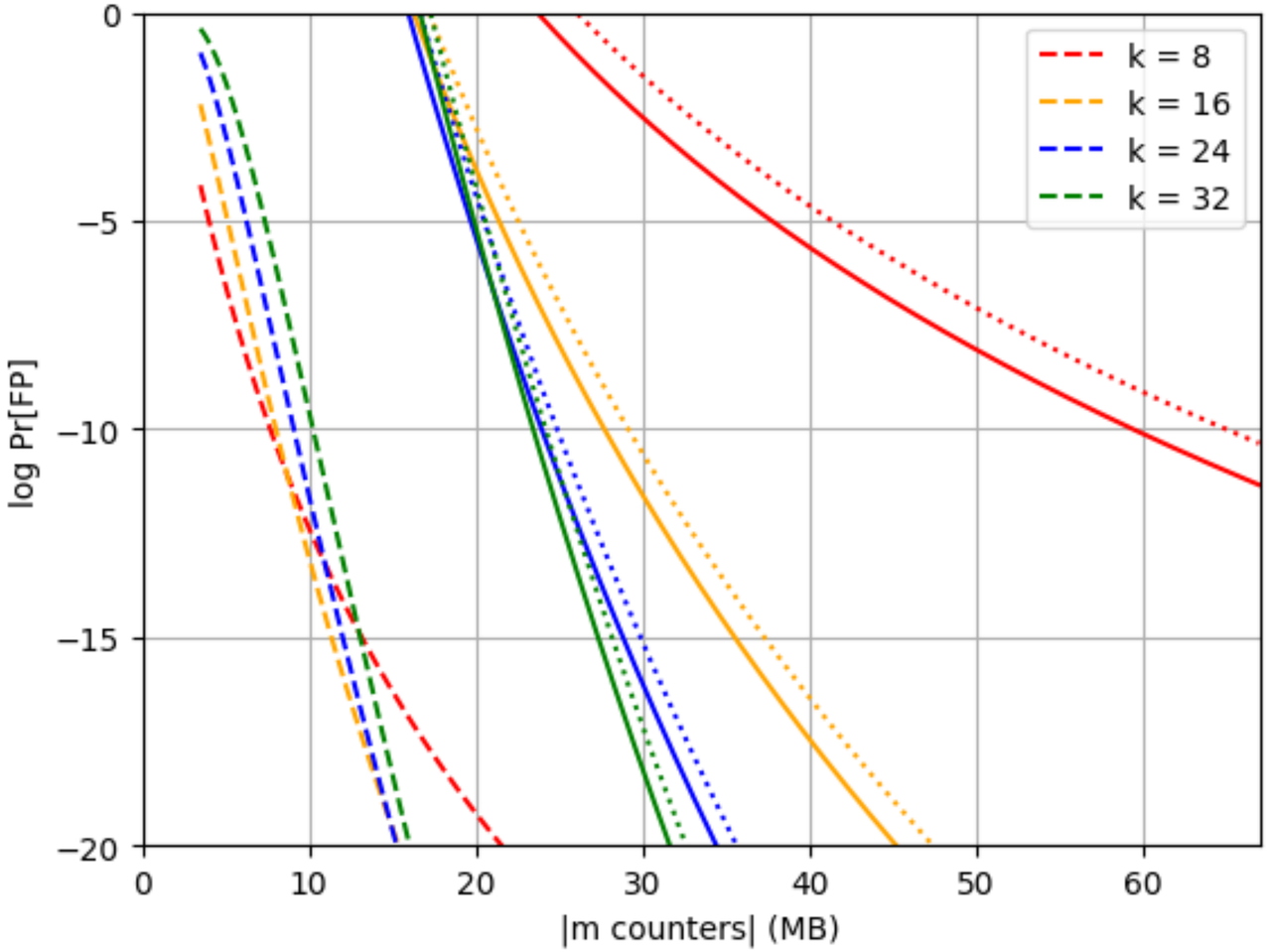
..... adversarial setting w/ deletions

Maximum counter value is 15

User makes 2^{20} insertions, deletions and queries

$\varepsilon = 2^{-128}$

Securing Counting filters in practice



------- honest setting

..... adversarial setting w/ deletions

——— adversarial setting w/o deletions

Maximum counter value is 15

User makes 2^{20} insertions, deletions and queries

$\varepsilon = 2^{-128}$

Final remarks

Final remarks

- Our work formalises the honest ‘view’ for AMQ-PDS with deletions, which is distinct from the one for the insertion-only case ([FPUV22]).

Final remarks

- Our work formalises the honest ‘view’ for AMQ-PDS with deletions, which is distinct from the one for the insertion-only case ([FPUV22]).
- Deletions notably enhance adversarial power.

Final remarks

- Our work formalises the honest ‘view’ for AMQ-PDS with deletions, which is distinct from the one for the insertion-only case ([FPUV22]).
- Deletions notably enhance adversarial power.
- However, with proper parameter selection, claimed correctness under adversaries remains achievable even in the presence of deletions.

Future works

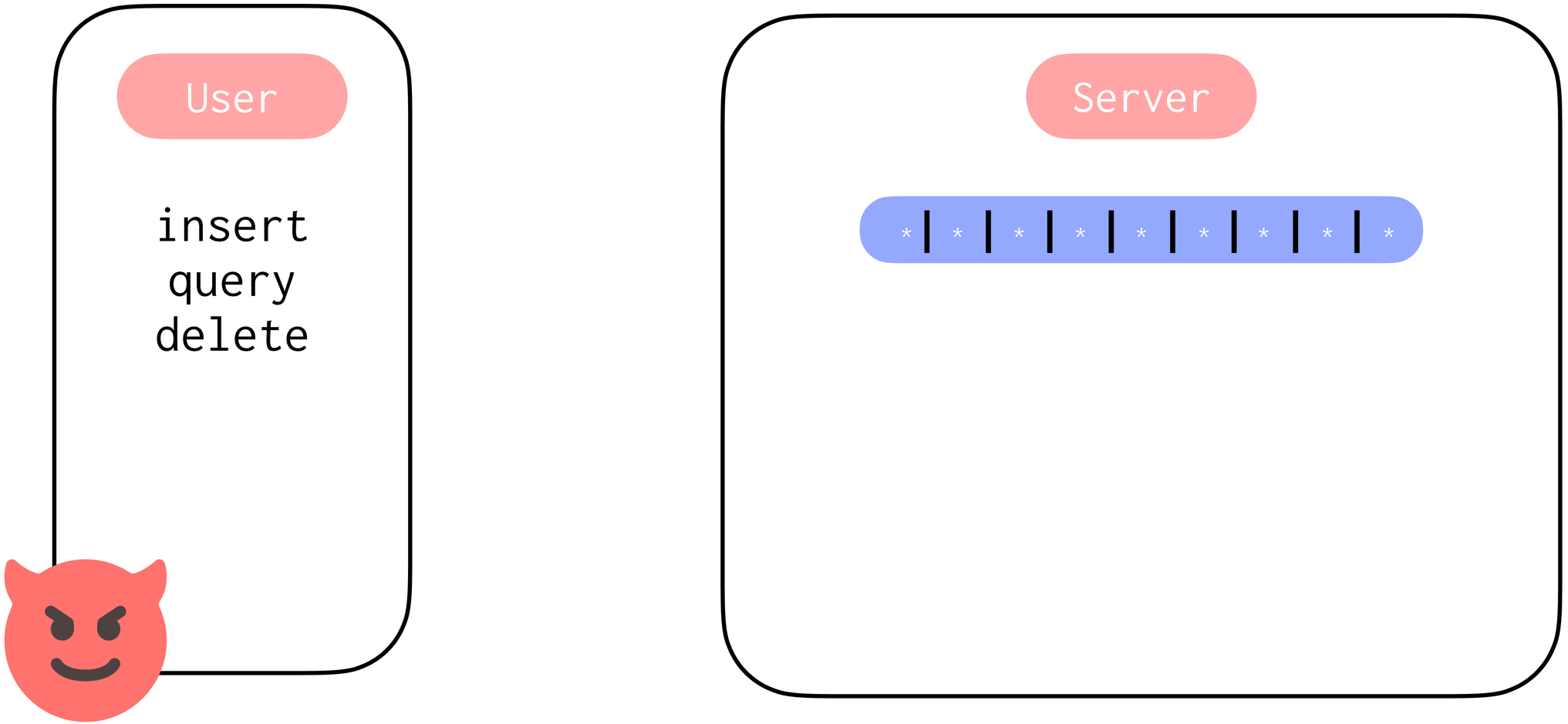
Future works

- Privacy of AMQ-PDS w/ deletions (see [FPUV22] for w/o deletions)

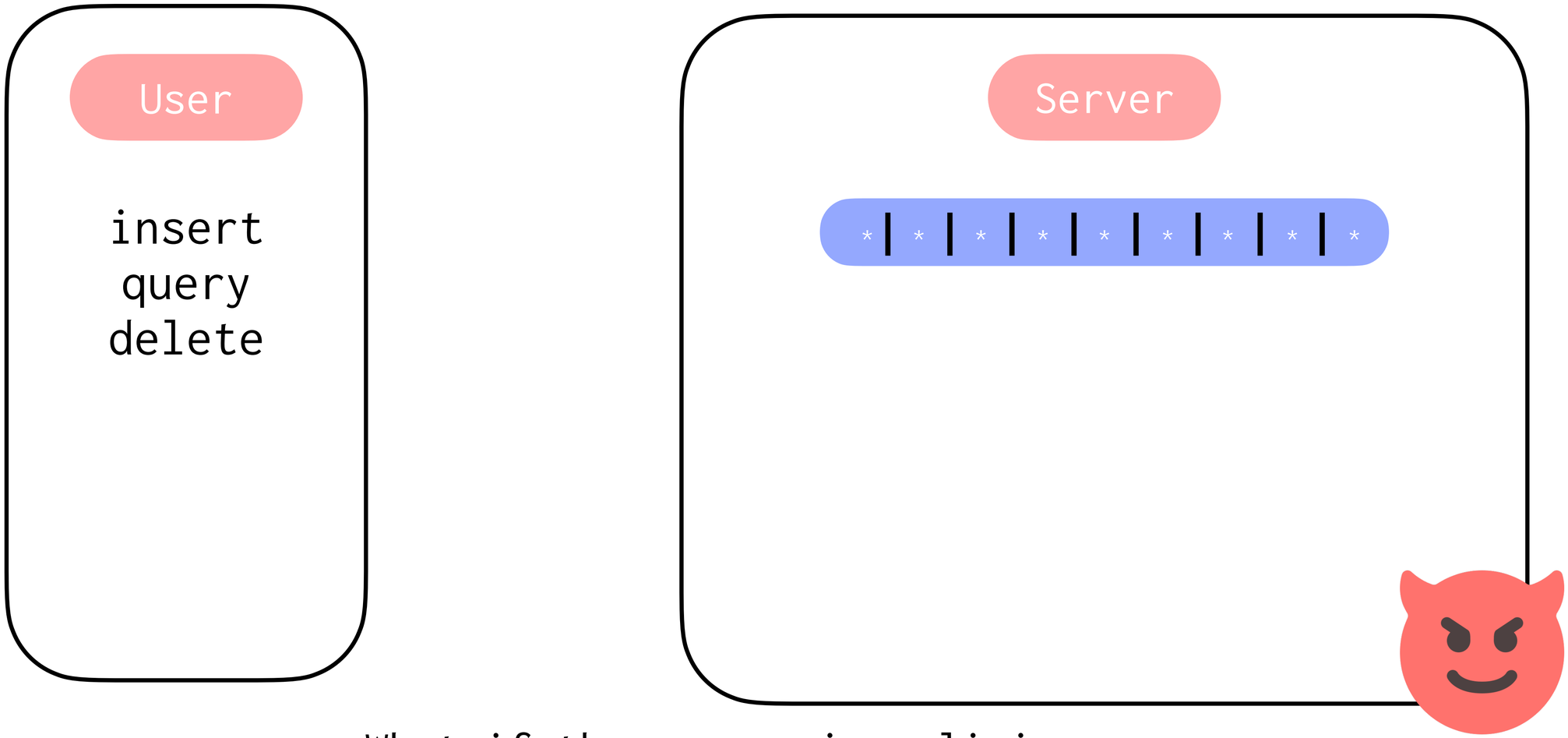
Future works

- Privacy of AMQ-PDS w/ deletions (see [FPUV22] for w/o deletions)
- Consider other PDS in adversarial settings (see [MFS23], [CPS19] for frequancy estimation PDS and their adversarial correctness)

Future work



Future work



What if the server is malicious
and the user is honest?

Thank you!

Follow up/parallel work:

- Privacy implications of AMQ-based PQ TLS authentication (CoNEXT24)
  - https://dl.acm.org/doi/10.1145/3680121.3697813
- Probabilistic Data Structures in the Wild: A Security Analysis of Redis
  - https://eprint.iacr.org/2024/1312
- Scalable Probabilistic Data Structures in Adverserial Enviroments (Raguso, Masters project)

Thank you!

Full paper: https://eprint.iacr.org/2024/1911

References

- MFS23
  - https://eprint.iacr.org/2023/1366
- FPUV22
  - https://eprint.iacr.org/2022/1186
- PR22
  - https://eprint.iacr.org/2021/1139
- CPS19
  - https://eprint.iacr.org/2019/1221
- NY15
  - https://eprint.iacr.org/2015/543
- FN
  - https://blog.fleek.network/post/bloom-and-cuckoo-filters-for-cache-summarization/