Compact Frequency Estimators

in Adversarial Environments

Sam A. Markelon
University of Florida

Mia Filić
ETH Zürich

Thomas Shrimpton
University of Florida

Probabilistic Data Structures (PDS)

A way to

compactly represent (stream of) data

and

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

Probabilistic Data Structures (PDS)

A way to

compactly represent (stream of) data

and

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

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

Probabilistic Data Structures (PDS)

A way to

compactly represent (stream of) data

and

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

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

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

Probabilistic Data Structures (PDS)

A way to

compactly represent (stream of) data

and

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

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

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

- Cardinality estimation
  How many distinct elements in the set?
  HyperLogLog, KMV estimator

Compact Frequency Estimators (CFE) help us

find the most visited pages on a website

identify possible DoS threats (network-monitoring systems)

Compact Frequency Estimators (CFE) help us

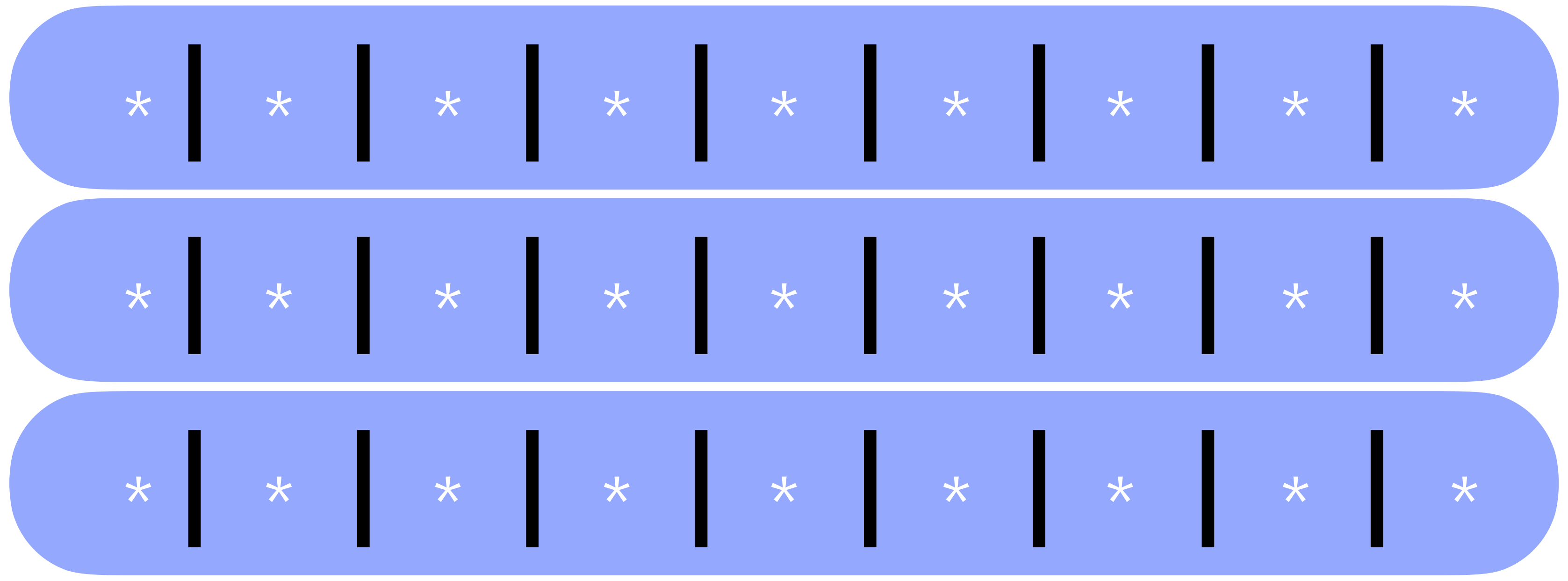
find the most visited pages on a website

identify possible DoS threats (network-monitoring systems)

Poseidon (Zhang et al. 2020)
Jaqen (Liu et al. 2021)
Ripple (Xing et al. 2021)

ACC-Turbo (Alcoz et al. 2022)
ALBUS (Scherrer et al. 2023)

Compact Frequency Estimators (CFE)



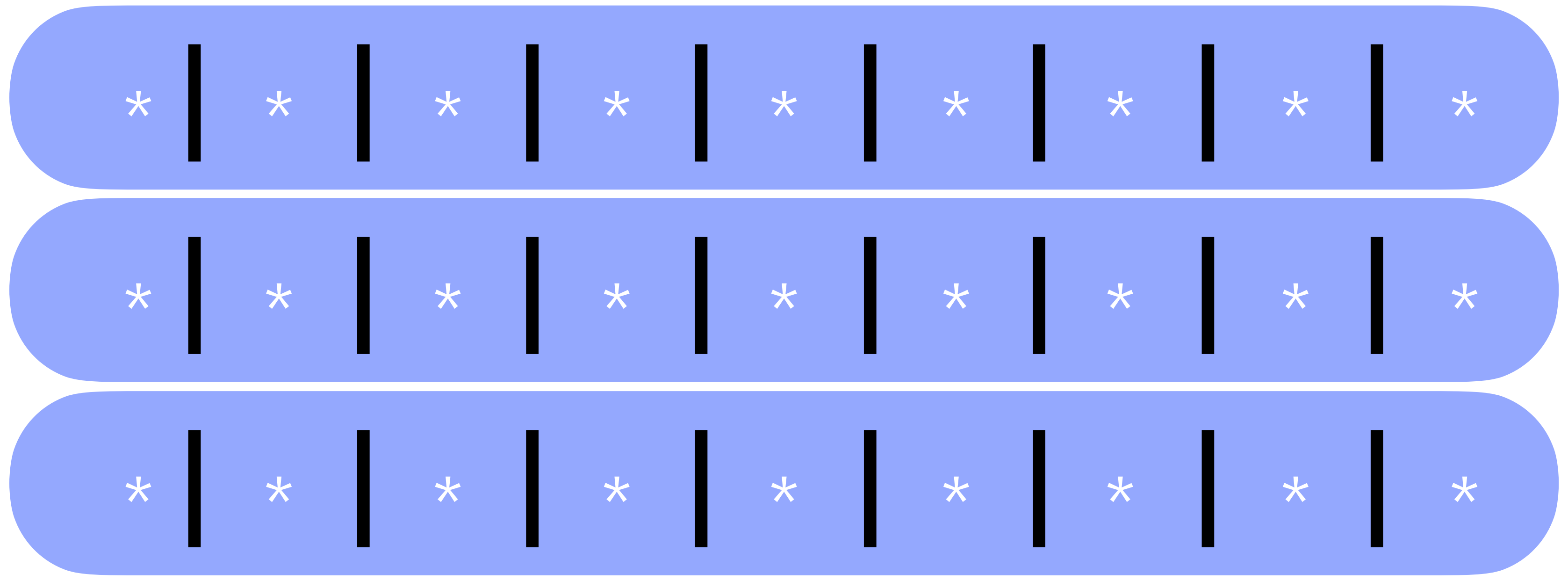
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

Compact Frequency Estimators (CFE)



CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

Compact Frequency Estimators (CFE)

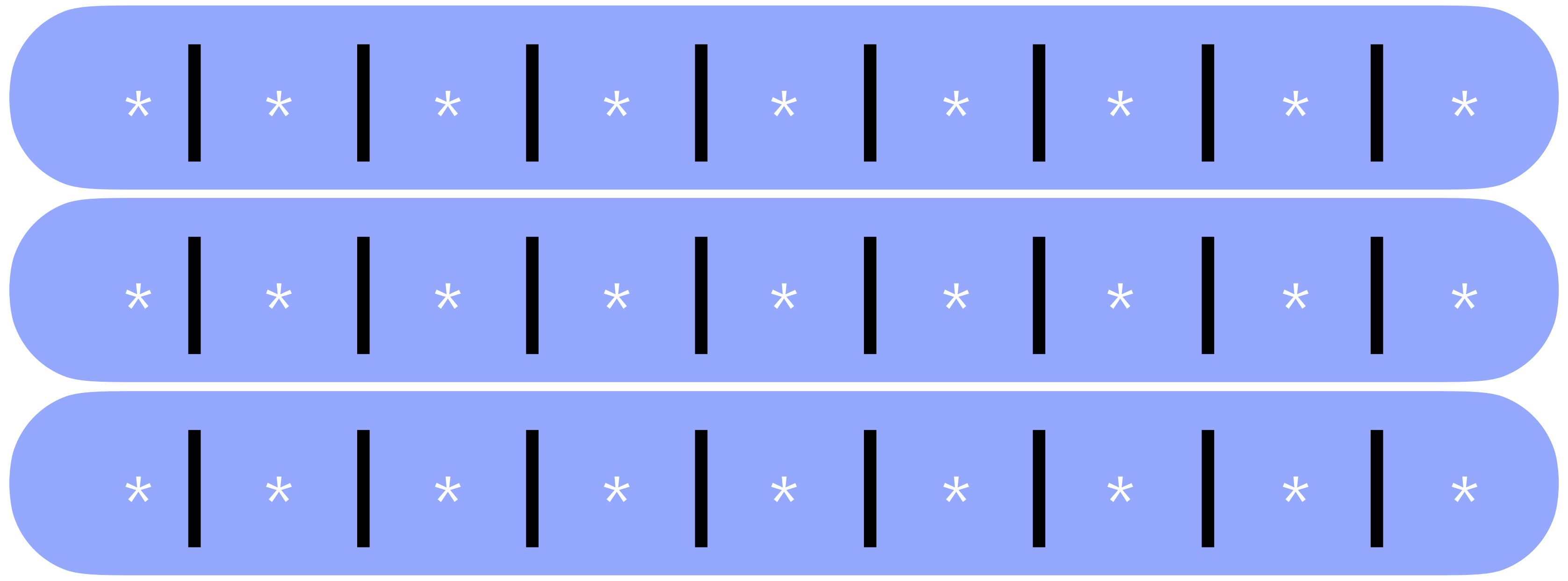
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

Compact Frequency Estimators (CFE)

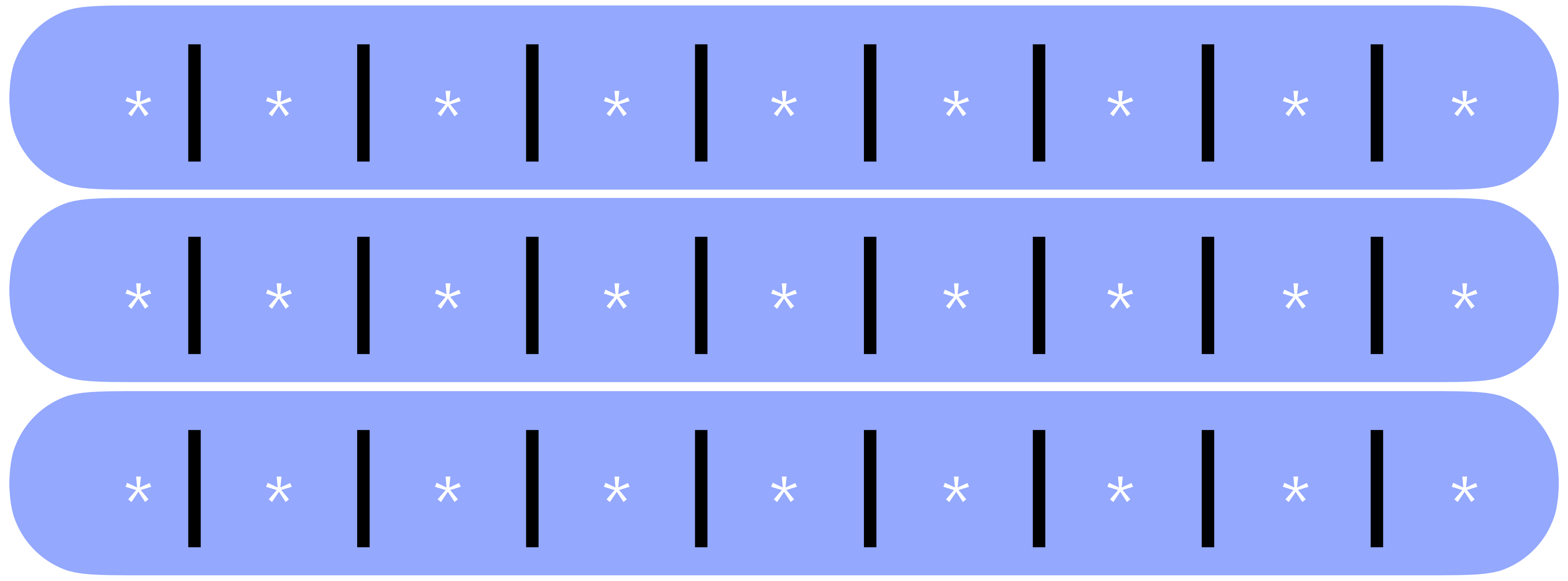


CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

Compact Frequency Estimators (CFE)



*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

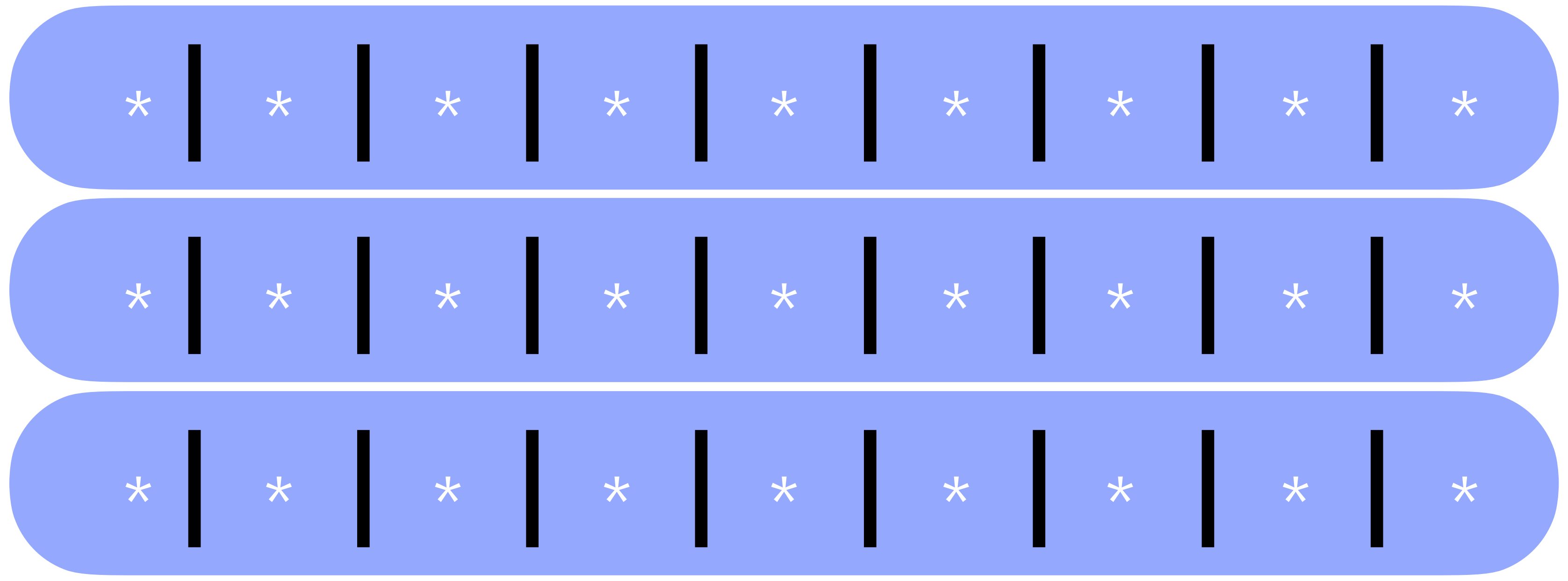
CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

?

Can CFE misbehave?



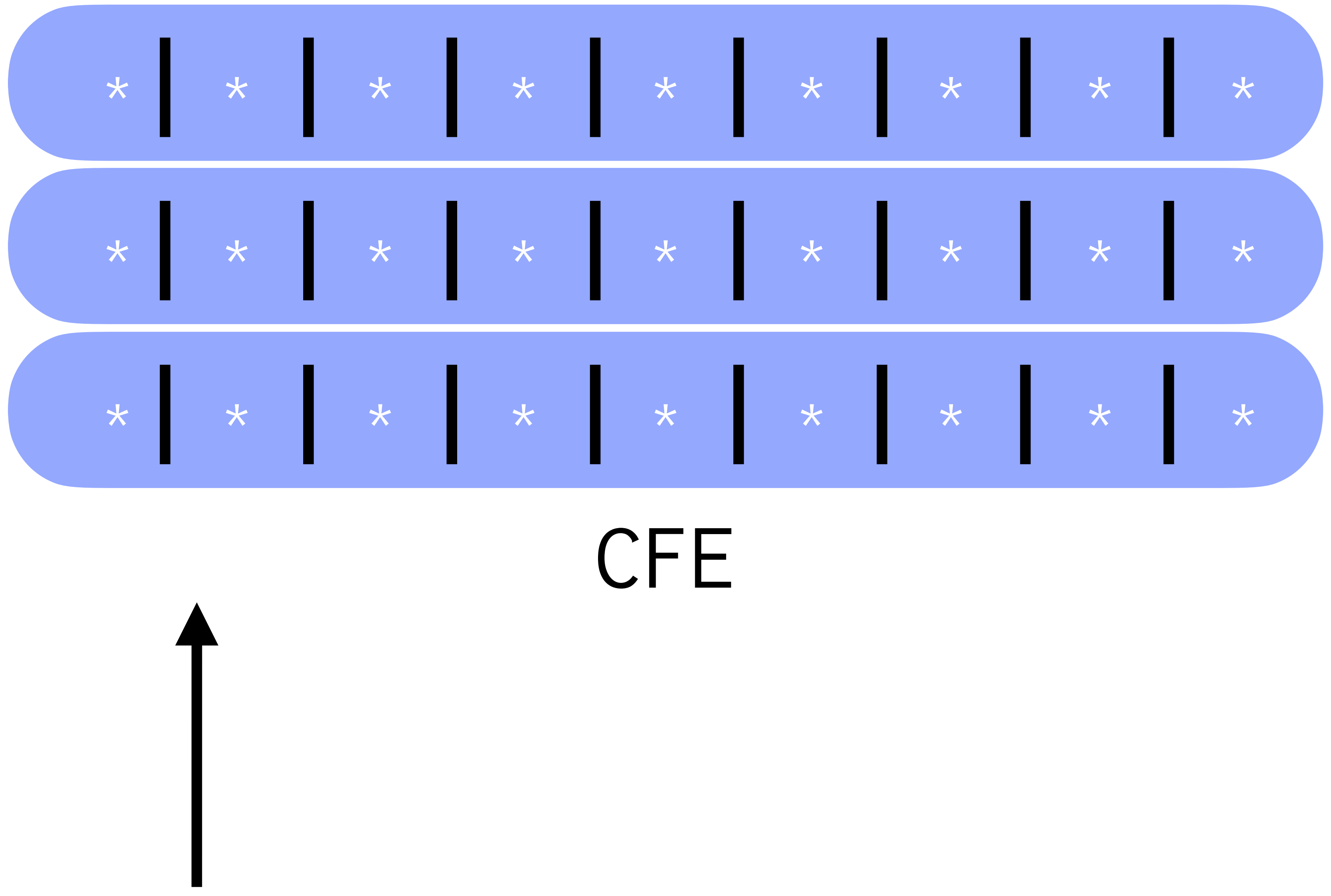
```
* | * | * | * | * | * | * | * | *
* | * | * | * | * | * | * | * | *
* | * | * | * | * | * | * | * | *
```

CFE

Stream

n,z,r,p,t,w,l,l,n,s,k,g,o,i,w,…

Can CFE misbehave?



*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

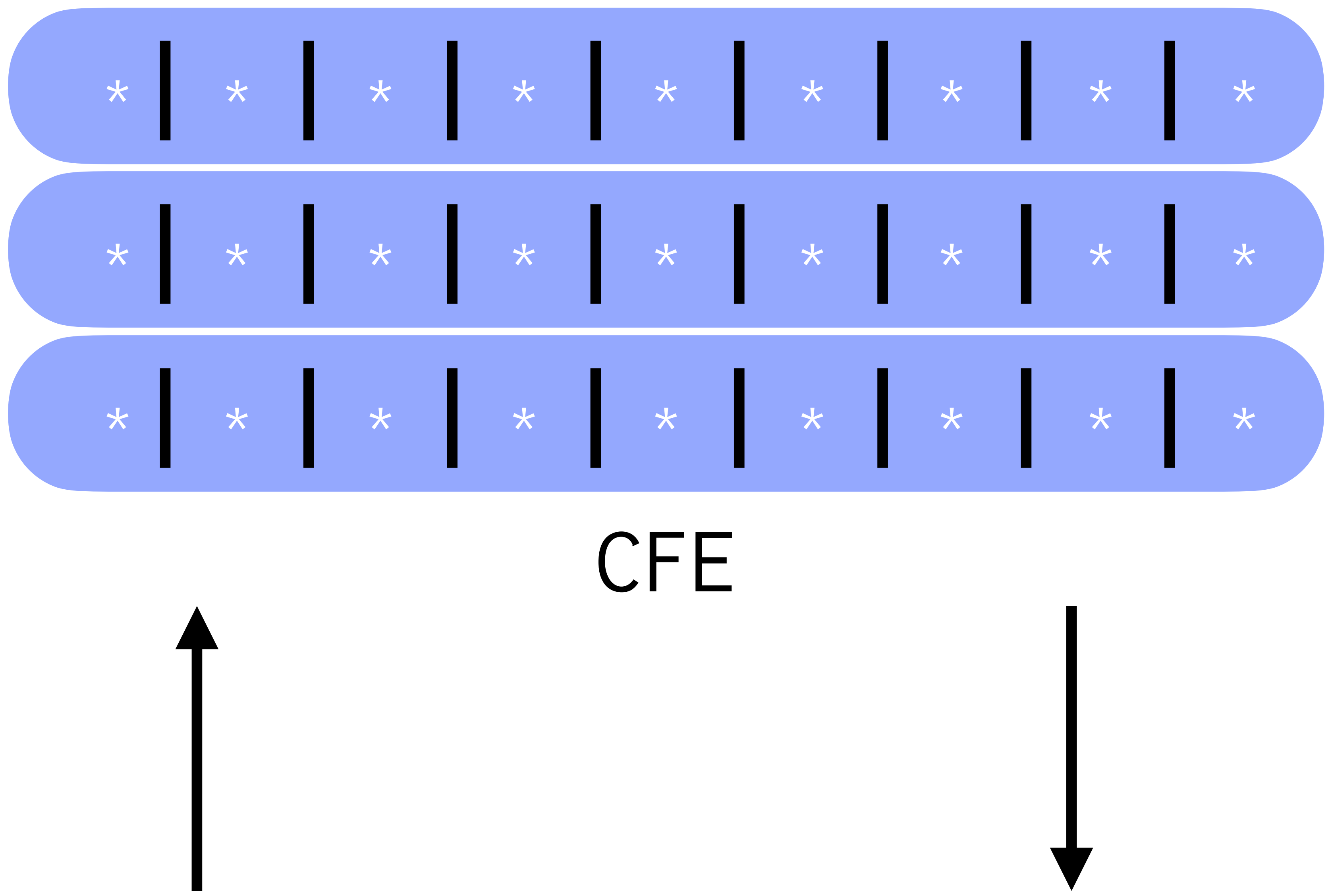
CFE

query(x)

Stream

n,z,r,p,t,w,l,l,n,s,k

Can CFE misbehave?



CFE

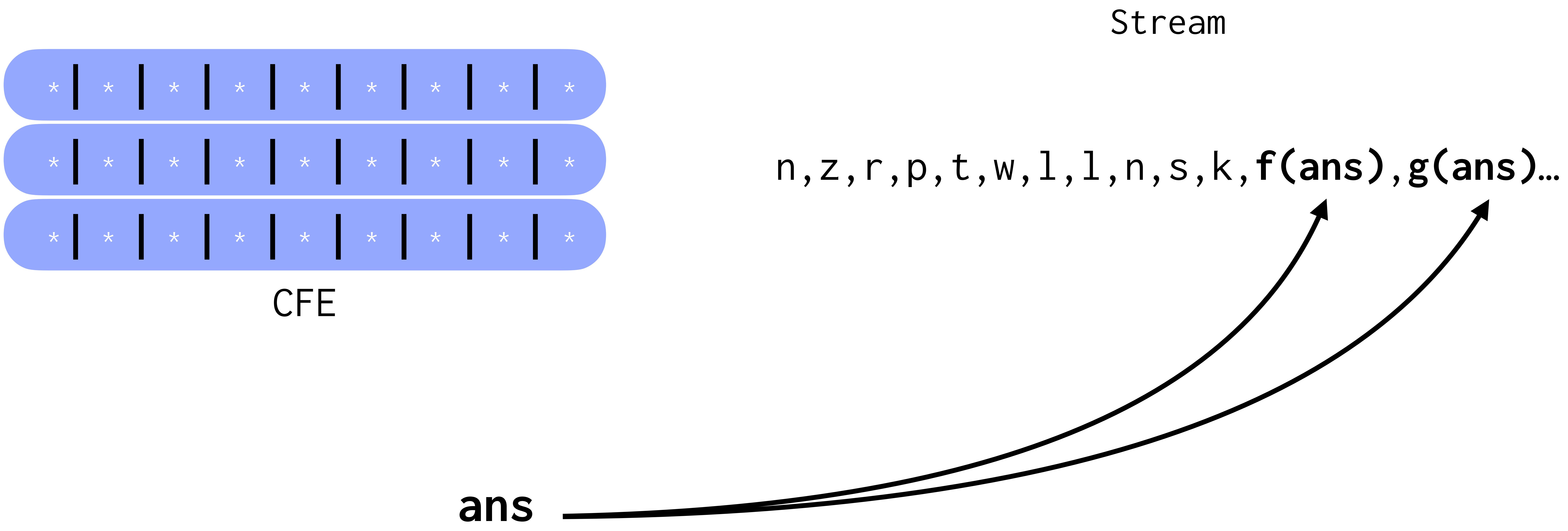
query(x)

ans

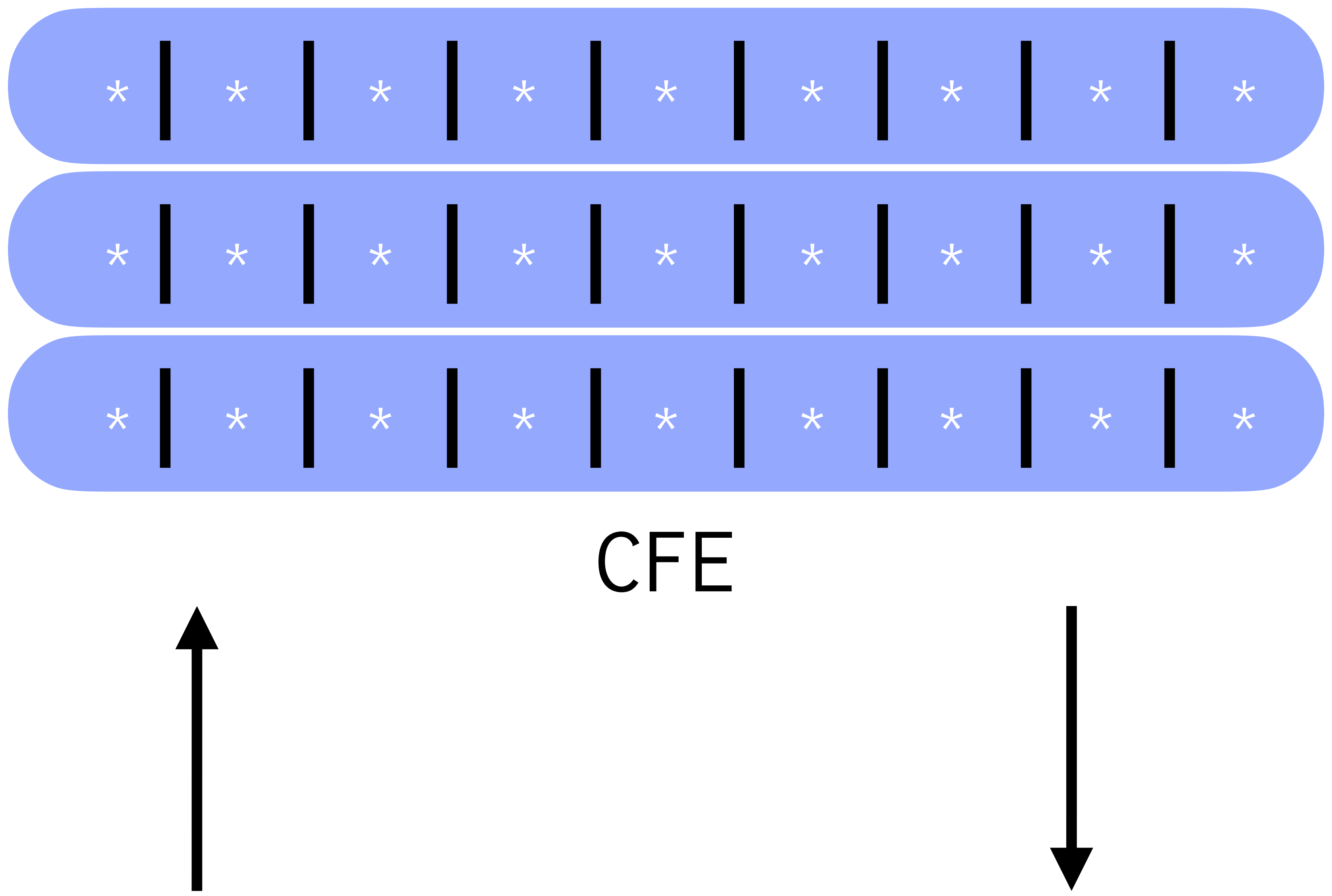
Stream

n,z,r,p,t,w,l,l,n,s,k

Can CFE misbehave?



Can CFE misbehave?



CFE

query(x)

** >> freq(x)
or
** << freq(x)

Stream

n,z,r,p,t,w,l,l,n,s,k,**g,o**,i,w,…

Our focus

Adversarial correctness of Compact Frequency Estimators (CFE)

- How does an adversary **interfere** with the functionality of Count-min sketch (CMS) and Heavy-keeper?

Our focus

Adversarial correctness of Compact Frequency Estimators (CFE)

- How does an adversary **interfere** with the functionality of Count-min sketch (CMS) and Heavy-keeper?

Exploration of a more robust CFE

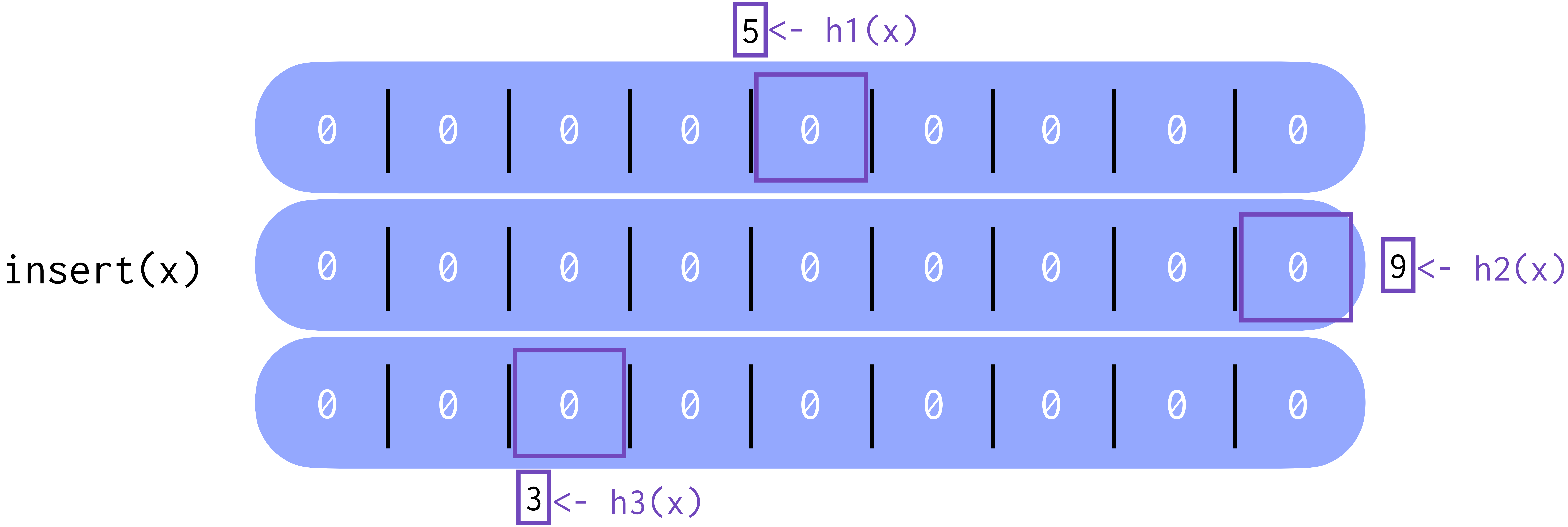
- How can we reduce estimation error and make CFE more robust in adversarial settings?

Count-min sketch (CMS)

m columns

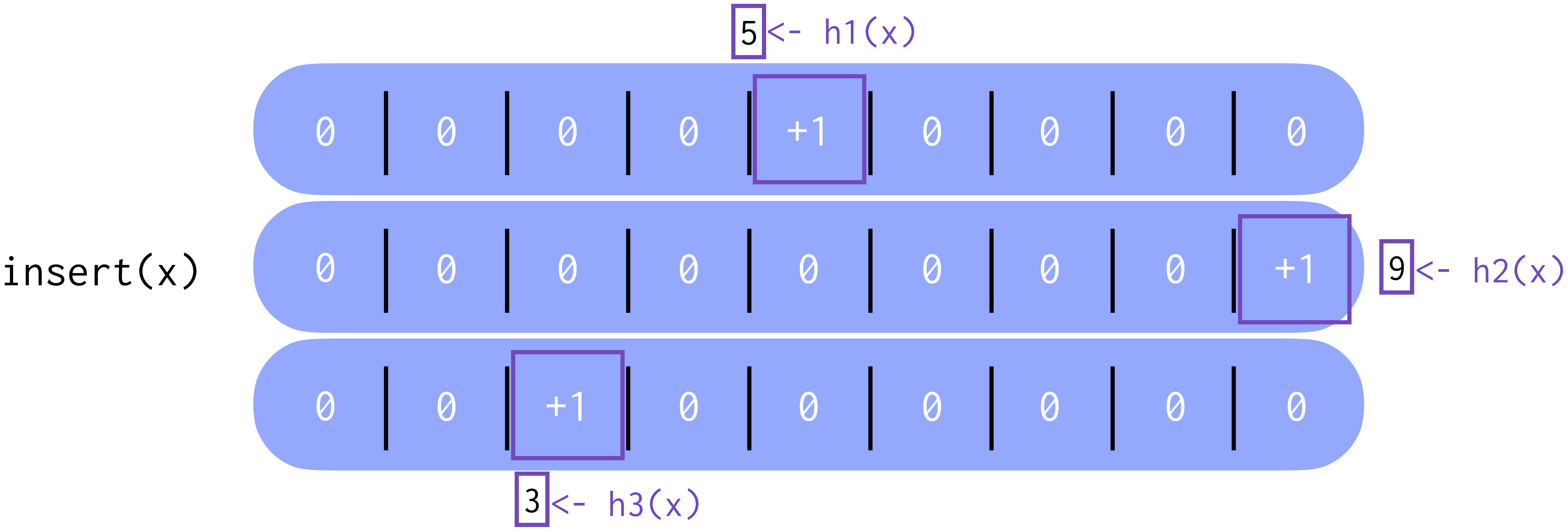
	0	0	0	0	0	0	0	0	0
k rows	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

CMS: how does it work?



0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

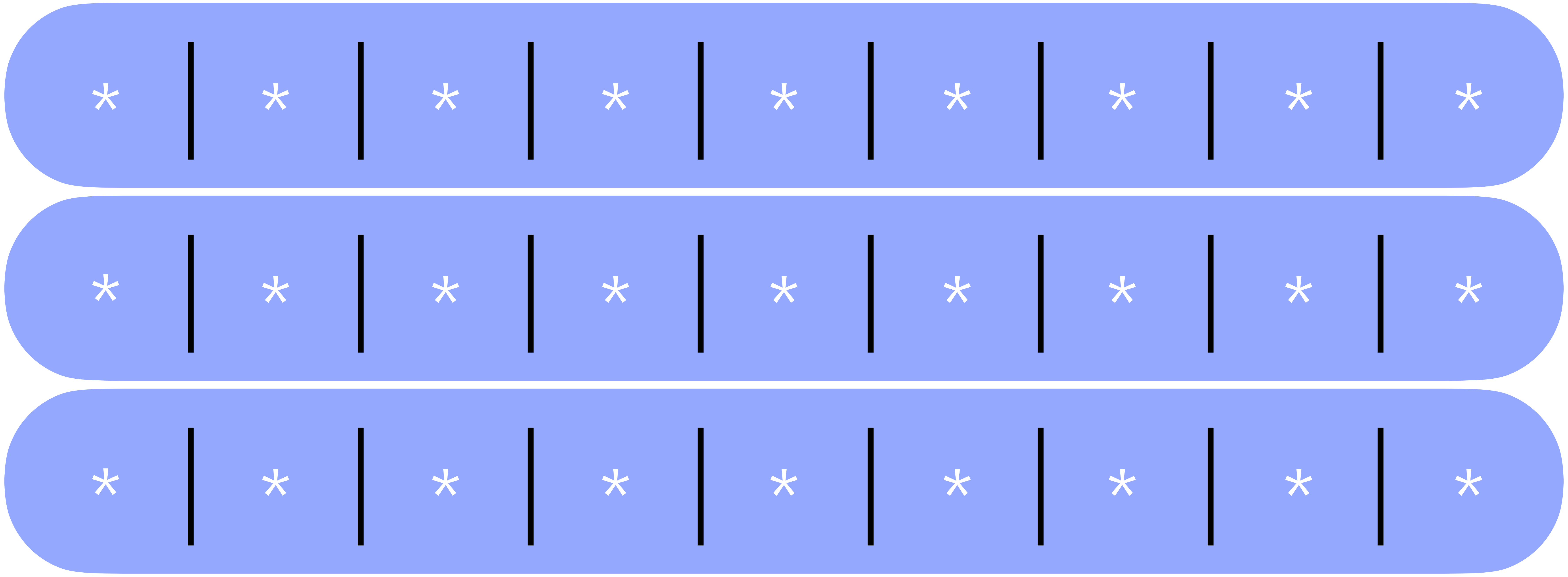
CMS: how does it work?



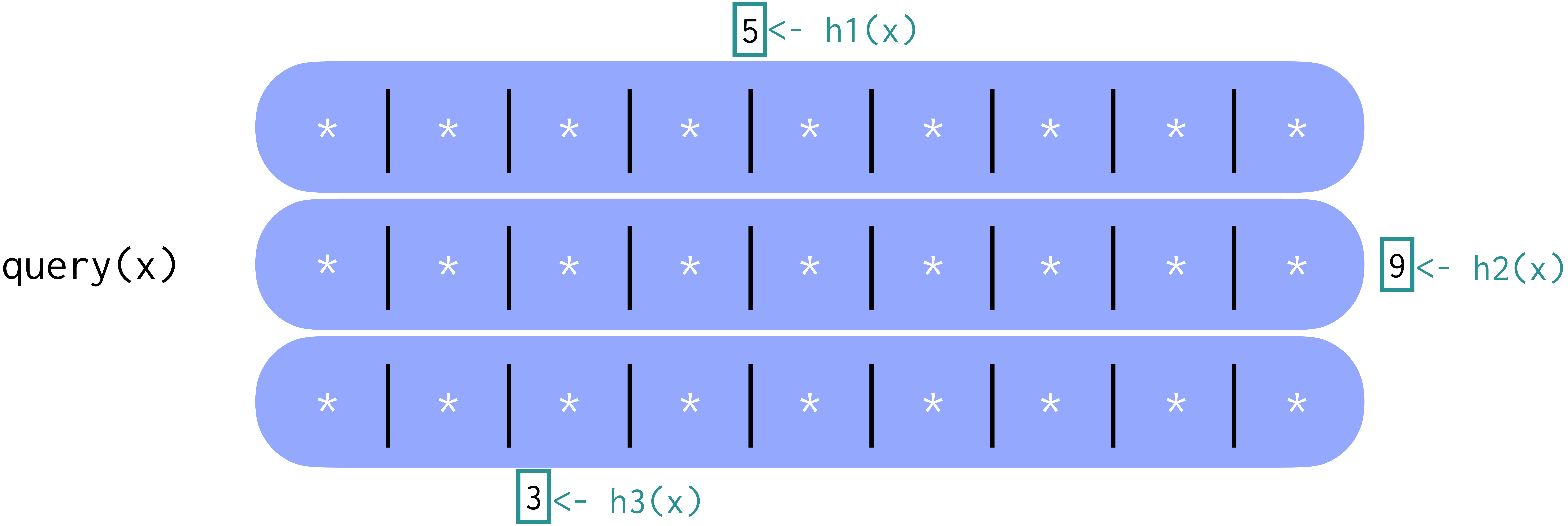
0	0	0	0	+1	0	0	0	0
0	0	0	0	0	0	0	0	+1
0	0	+1	0	0	0	0	0	0

CMS: how does it work?

query(x)



CMS: how does it work?



5 <- h1(x)

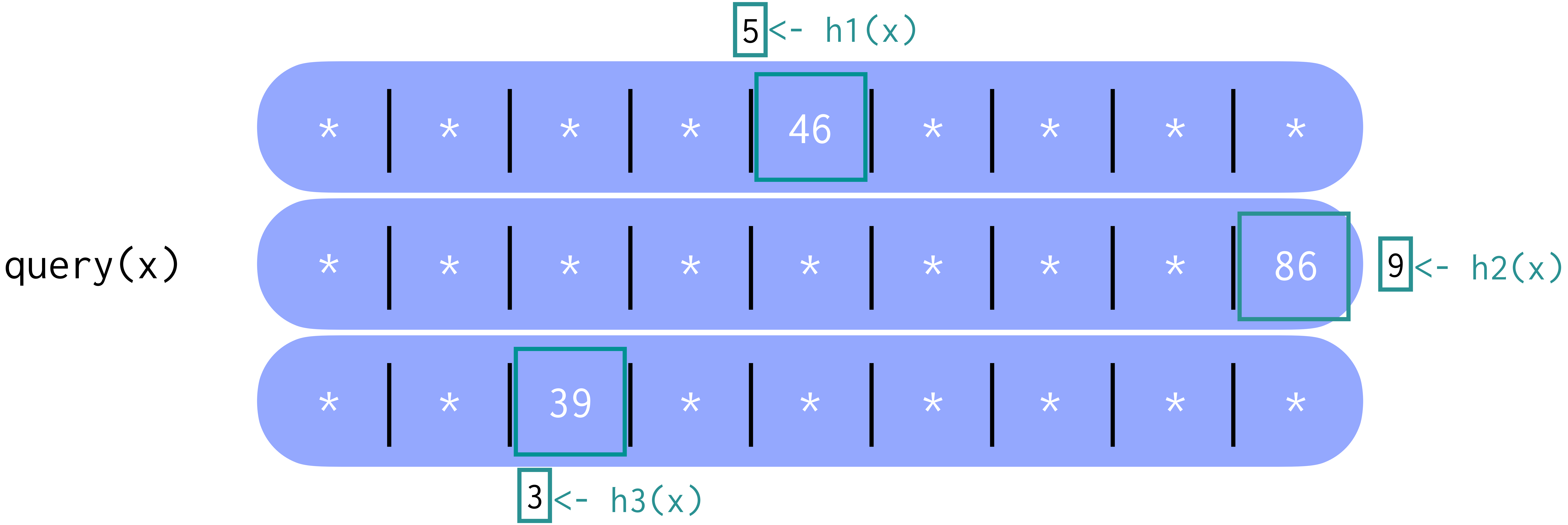
query(x)

*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

9 <- h2(x)

3 <- h3(x)

CMS: how does it work?



5 <- h1(x)

query(x)

* | * | * | * | 46 | * | * | * | *

* | * | * | * | * | * | * | * | 86

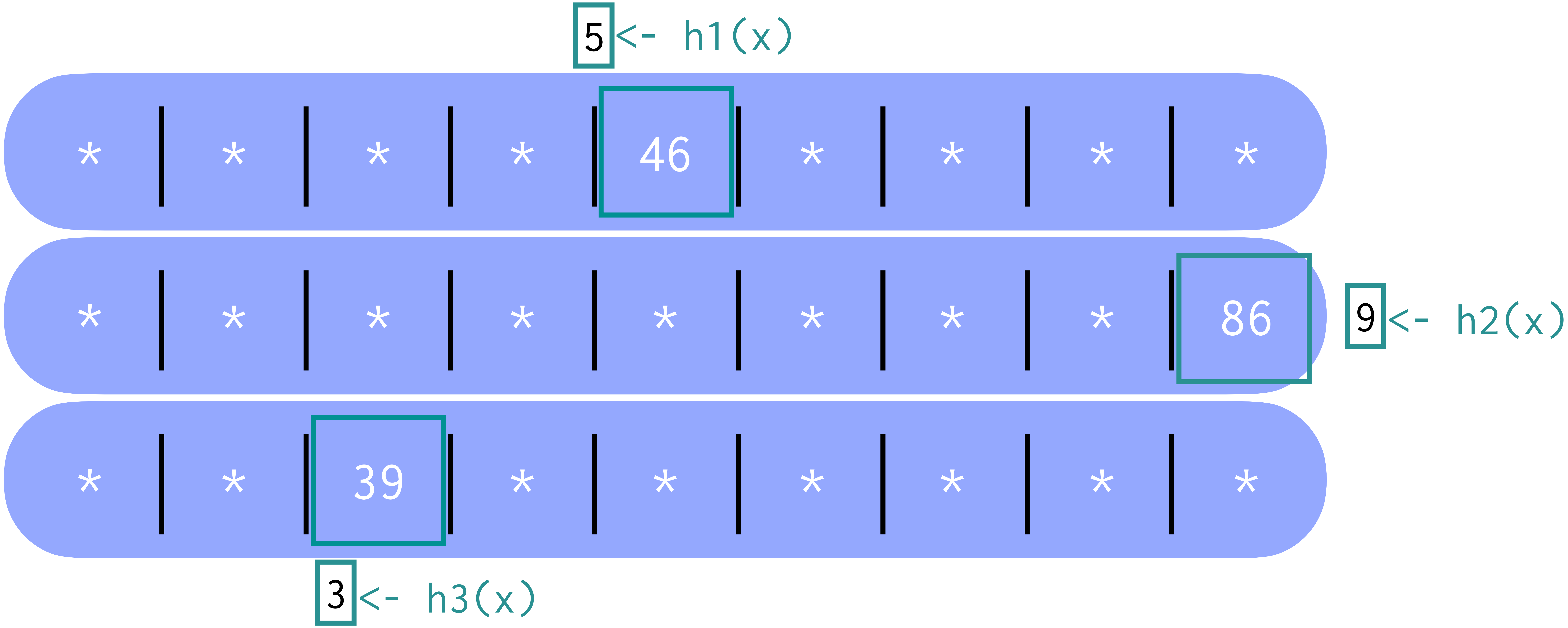
* | * | 39 | * | * | * | * | * | *

9 <- h2(x)

3 <- h3(x)

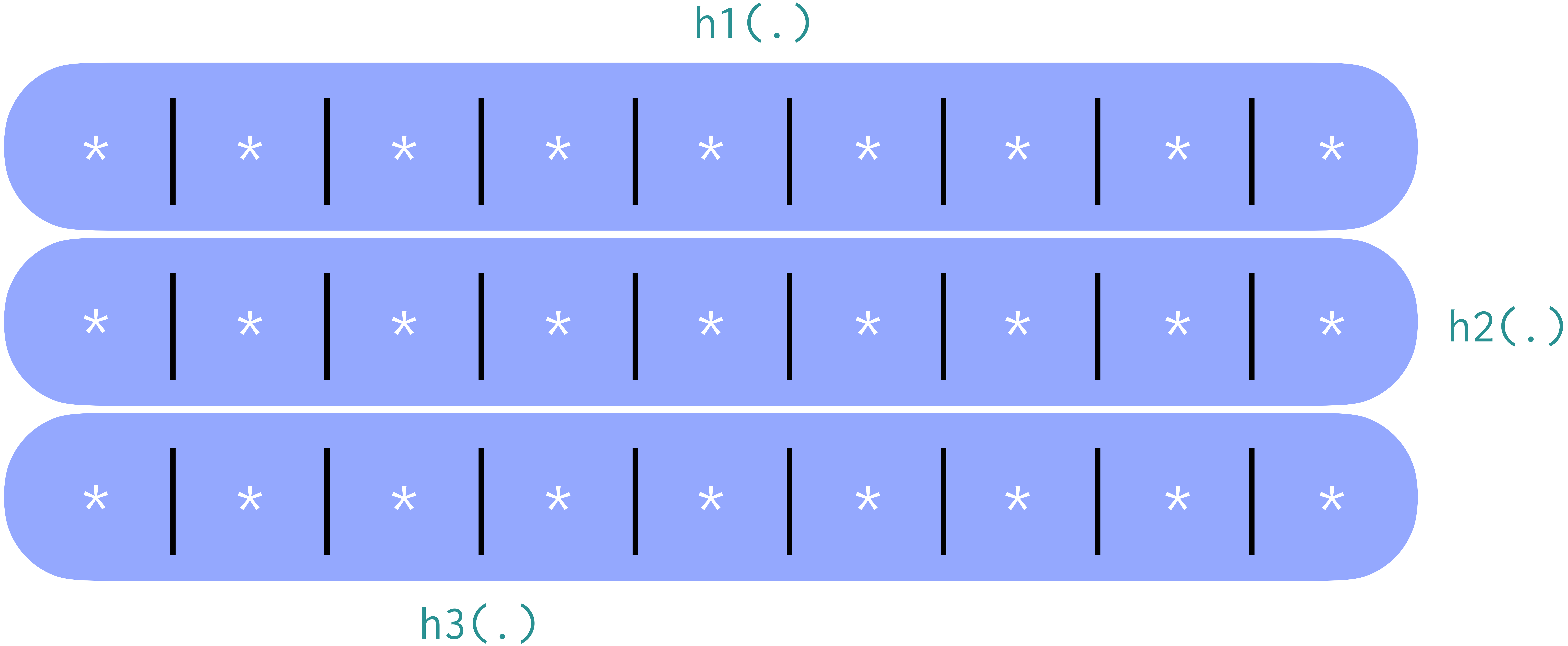
CMS: how does it work?

CMS(x)=39

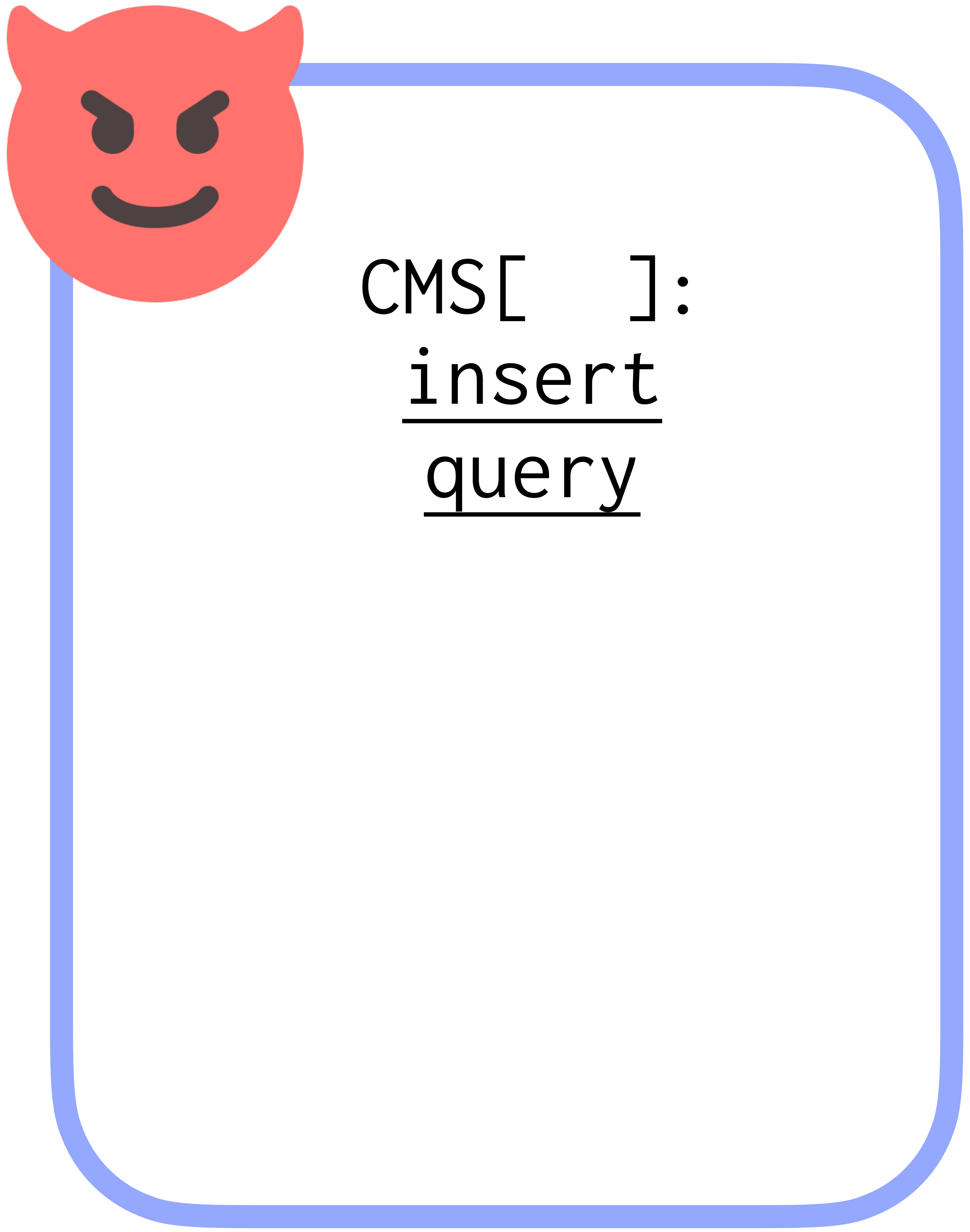


*	*	*	*	46	*	*	*	*
*	*	*	*	*	*	*	*	86
*	*	39	*	*	*	*	*	*

Count-min sketch (CMS)



CMS: attack model

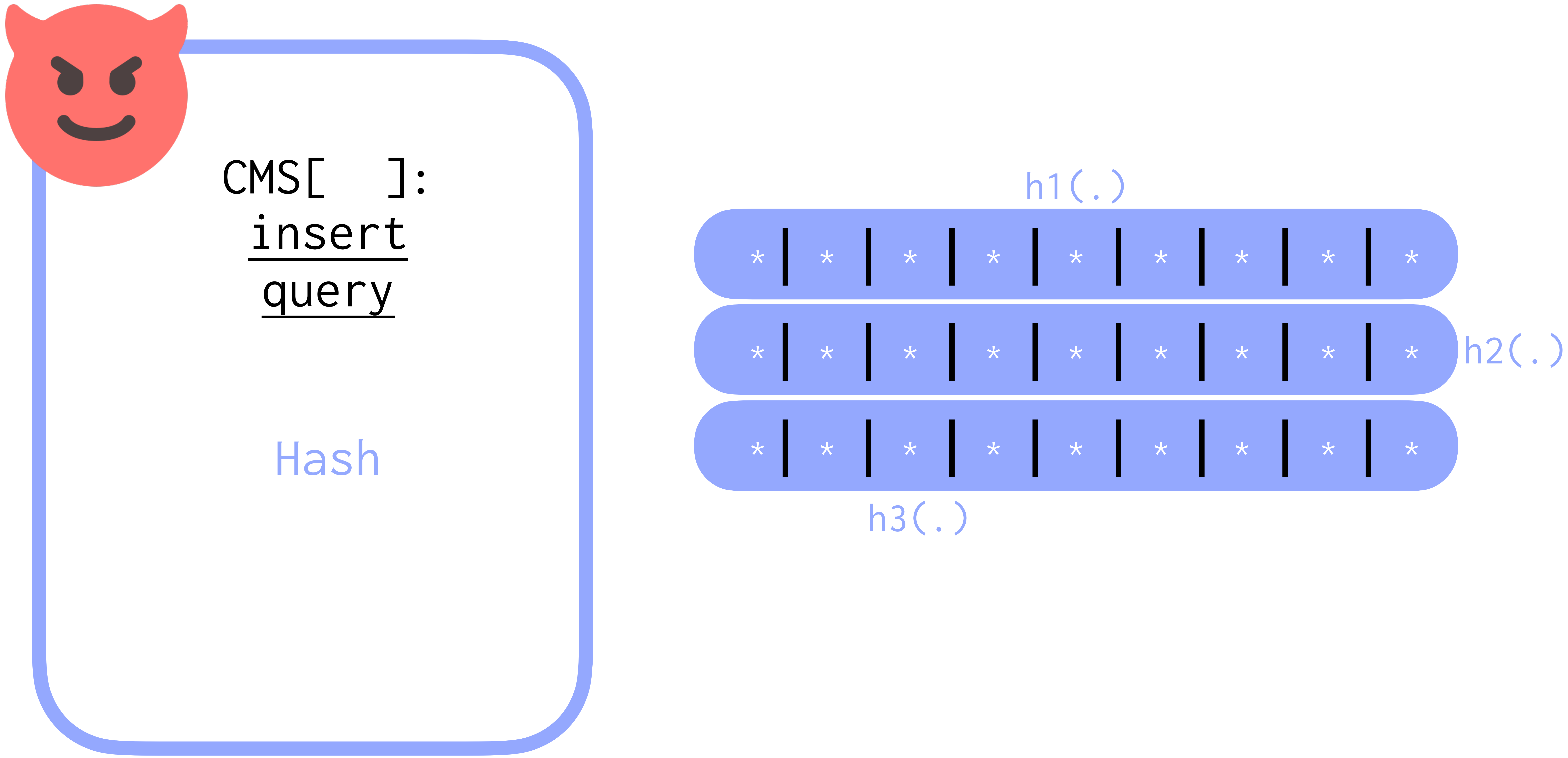


CMS[ ]:

insert

query

CMS: attack model



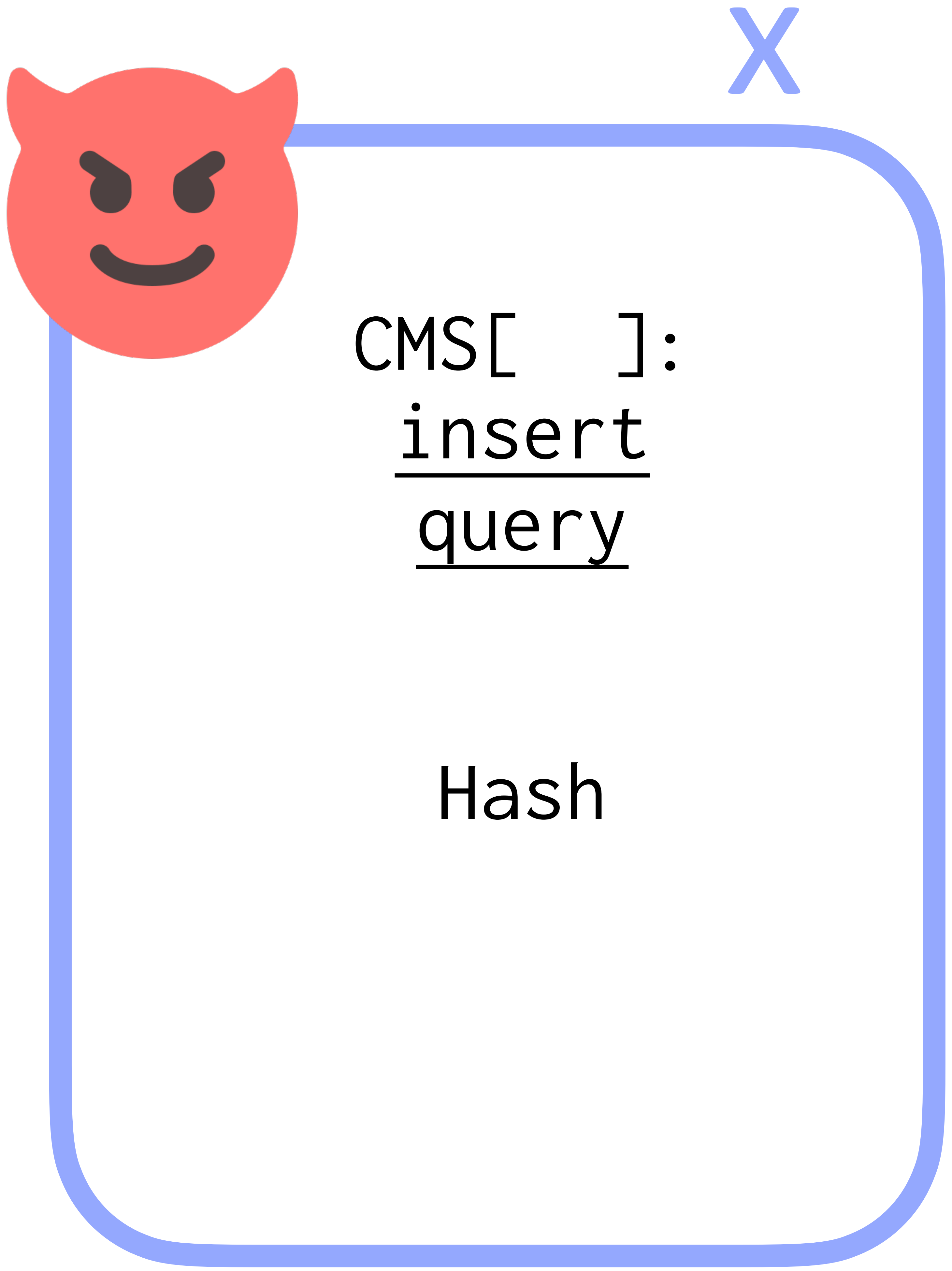
CMS[ ]:

insert

query

Hash

CMS: attack goal



X

CMS[ ]:

insert

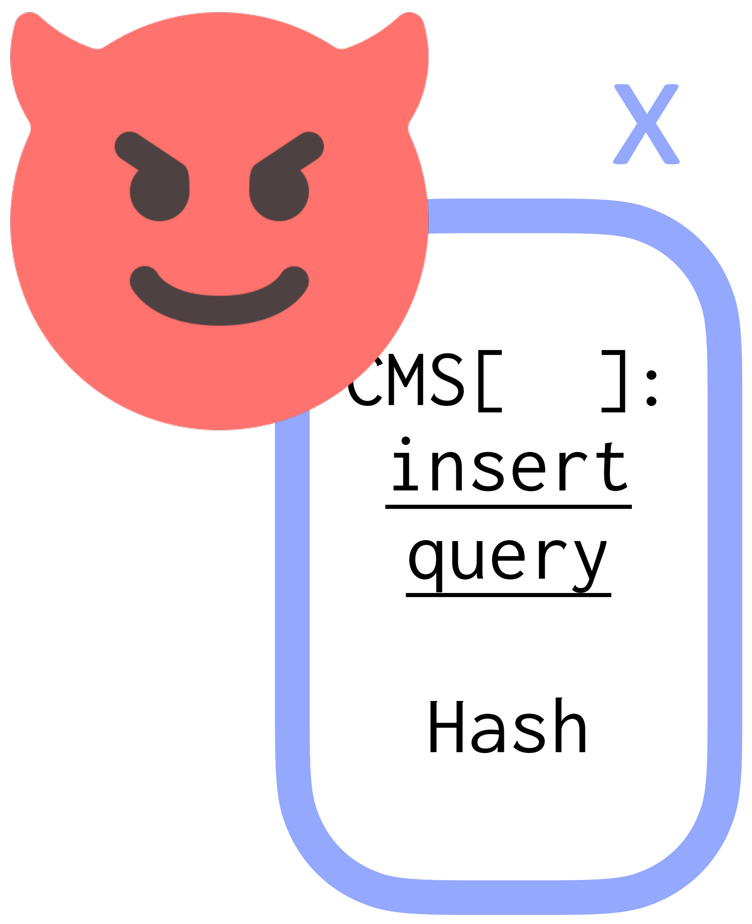
query

Hash

Maximise CMS error

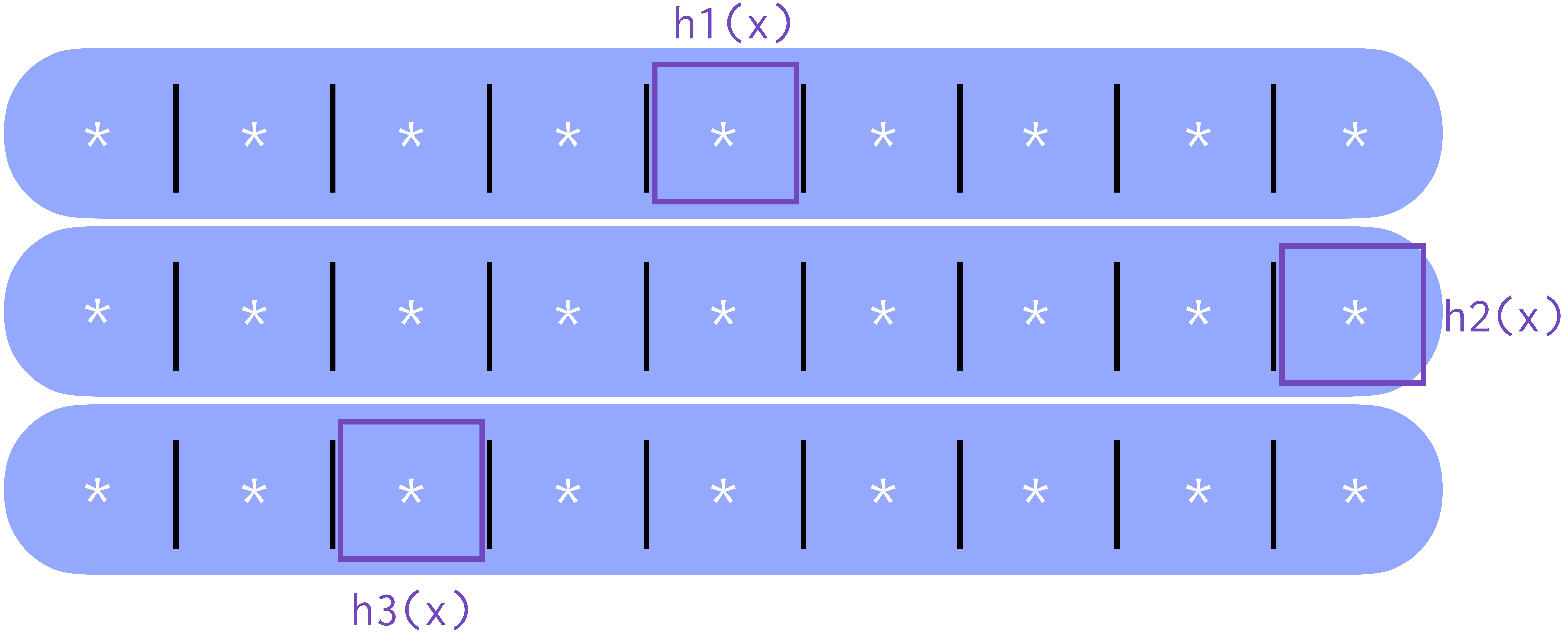
query(x) >> true_frequency(x)

CMS: attack

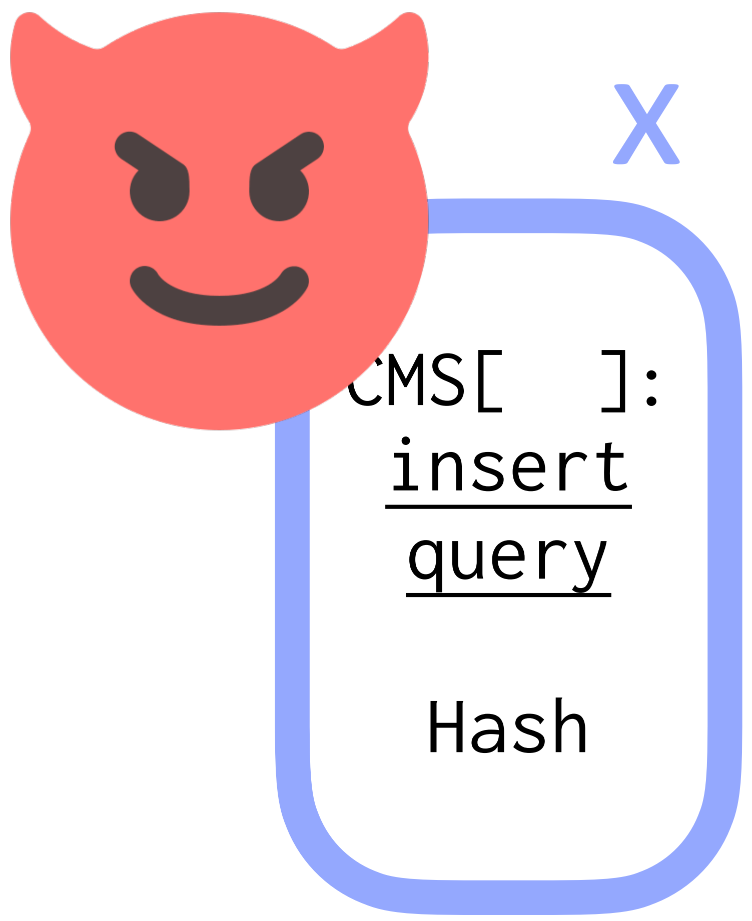


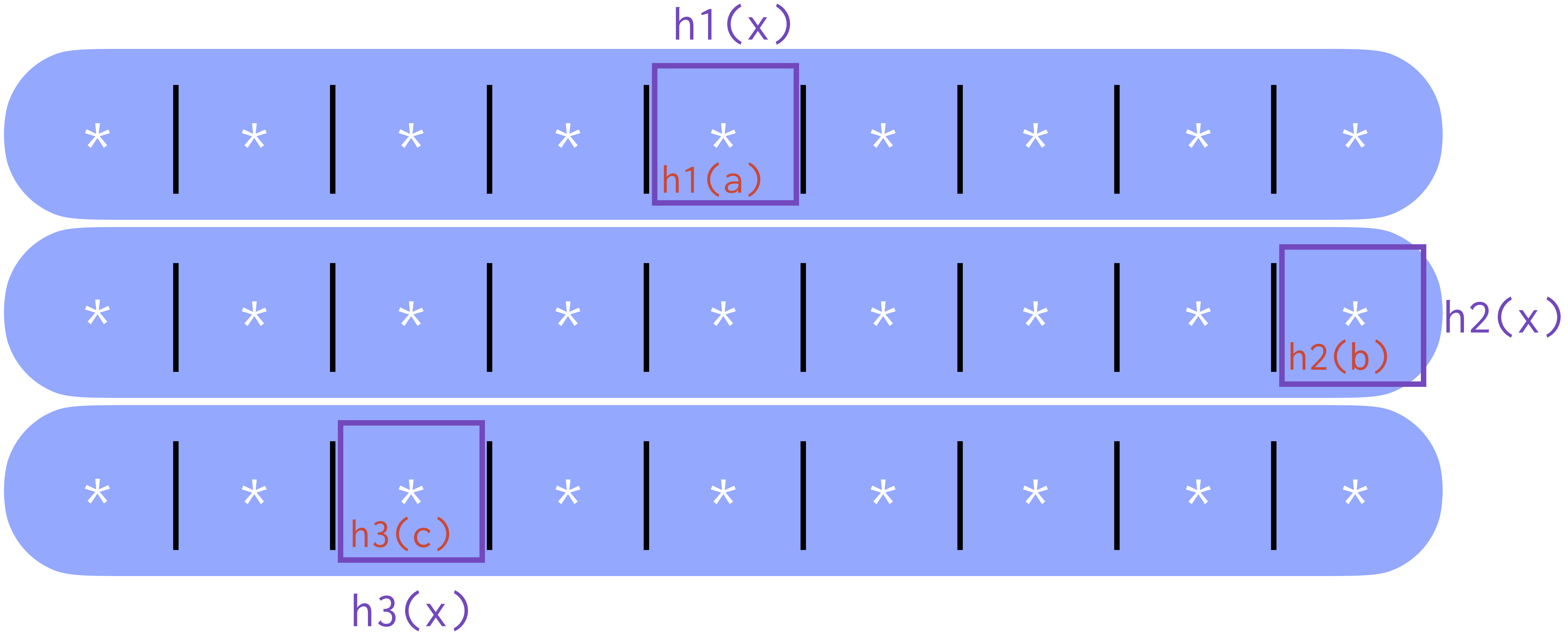
CMS[ ]:
insert
query

Hash

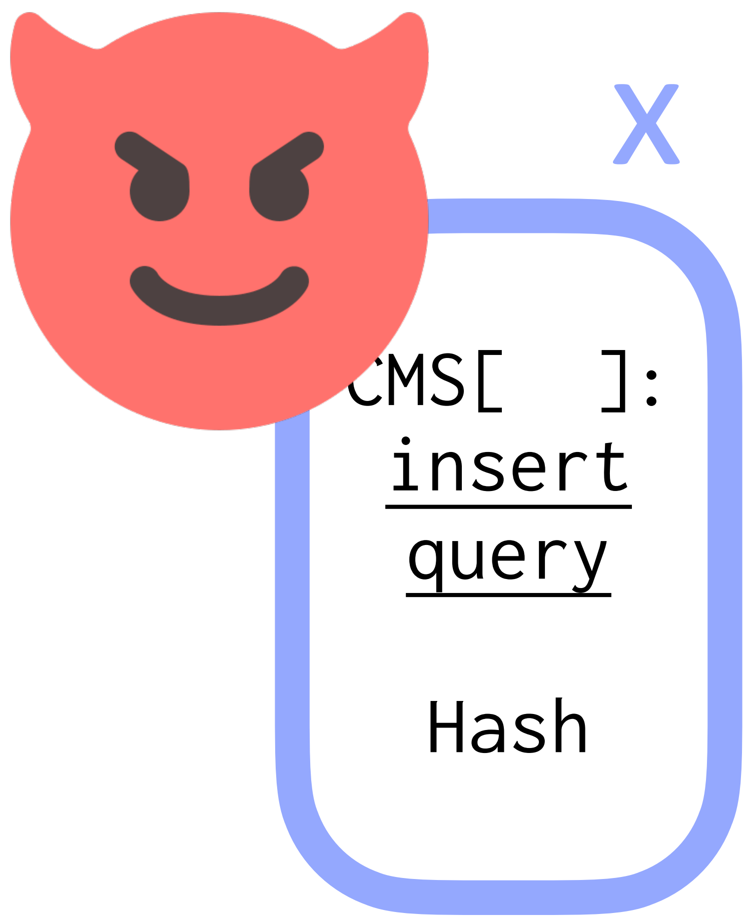


CMS: attack

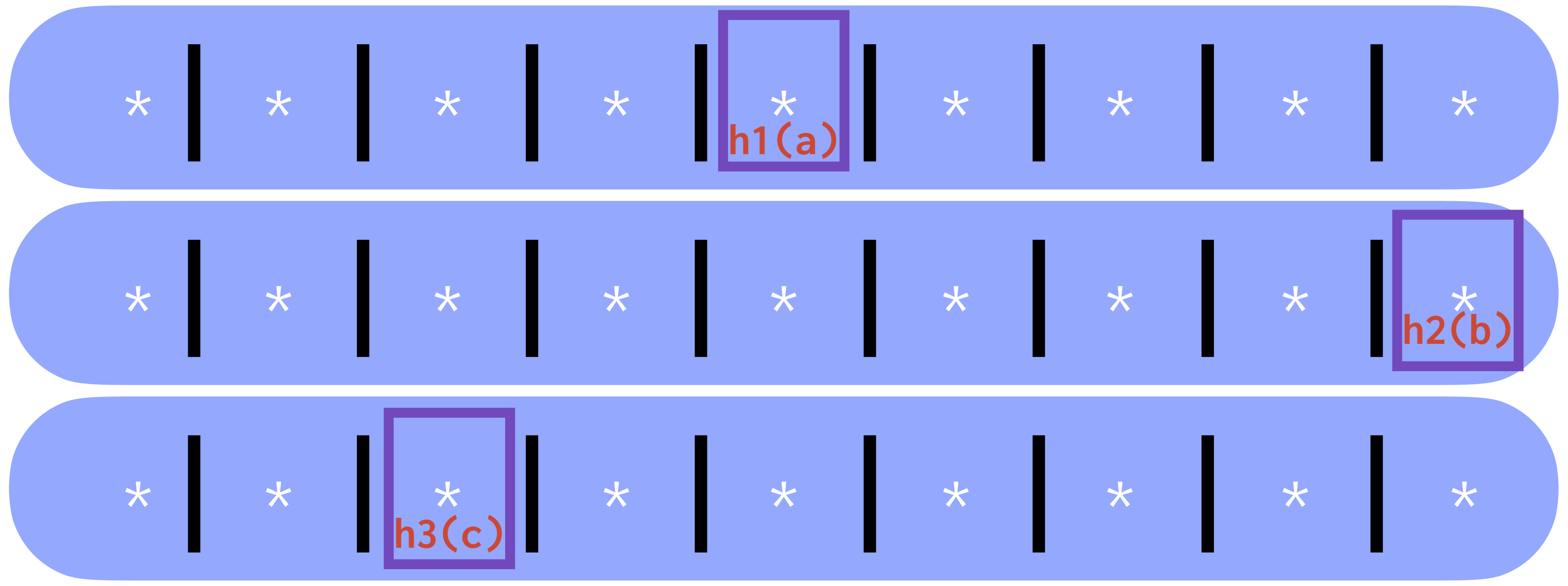




Cover set = {a, b, c}

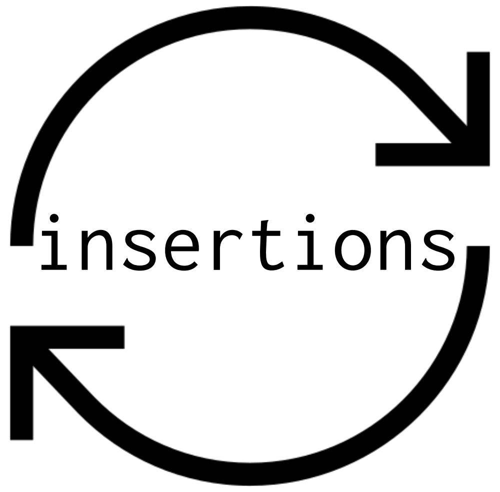


CMS: attack

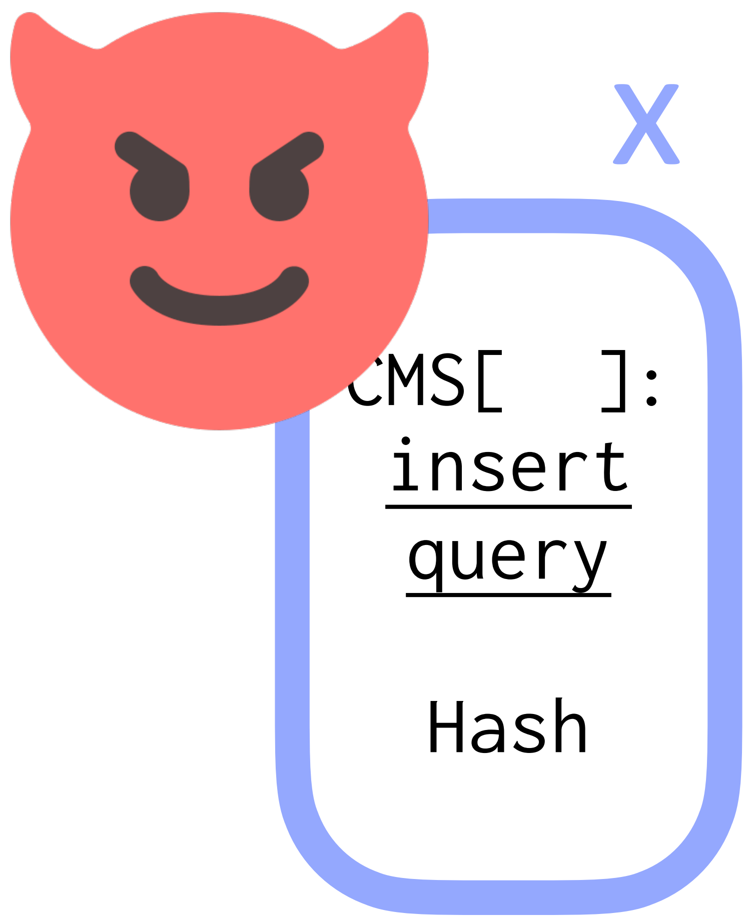


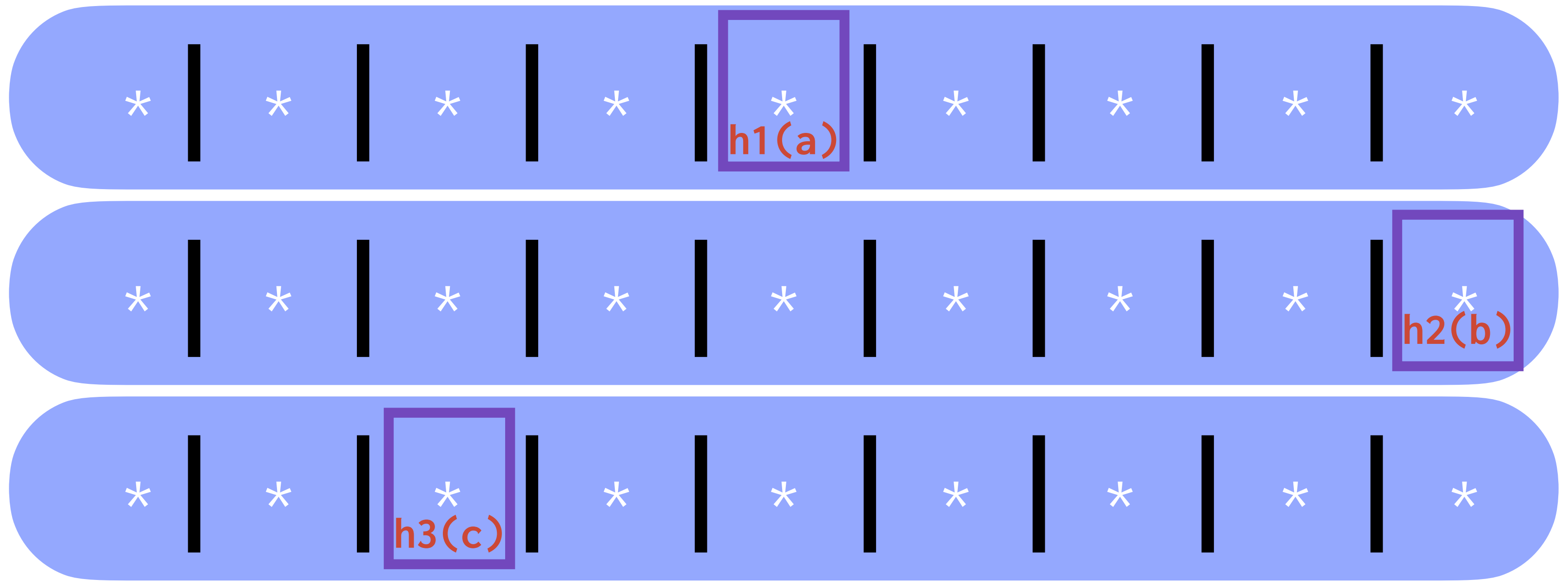
Cover set = {a, b, c}

Cover set

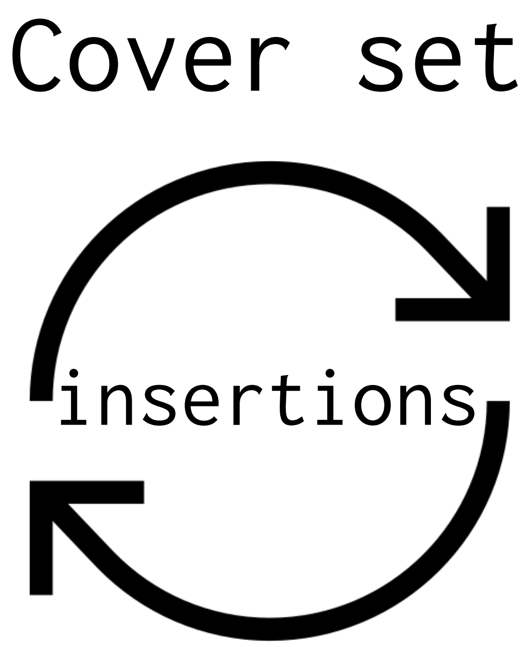


CMS: attack





Cover set = {a, b, c}



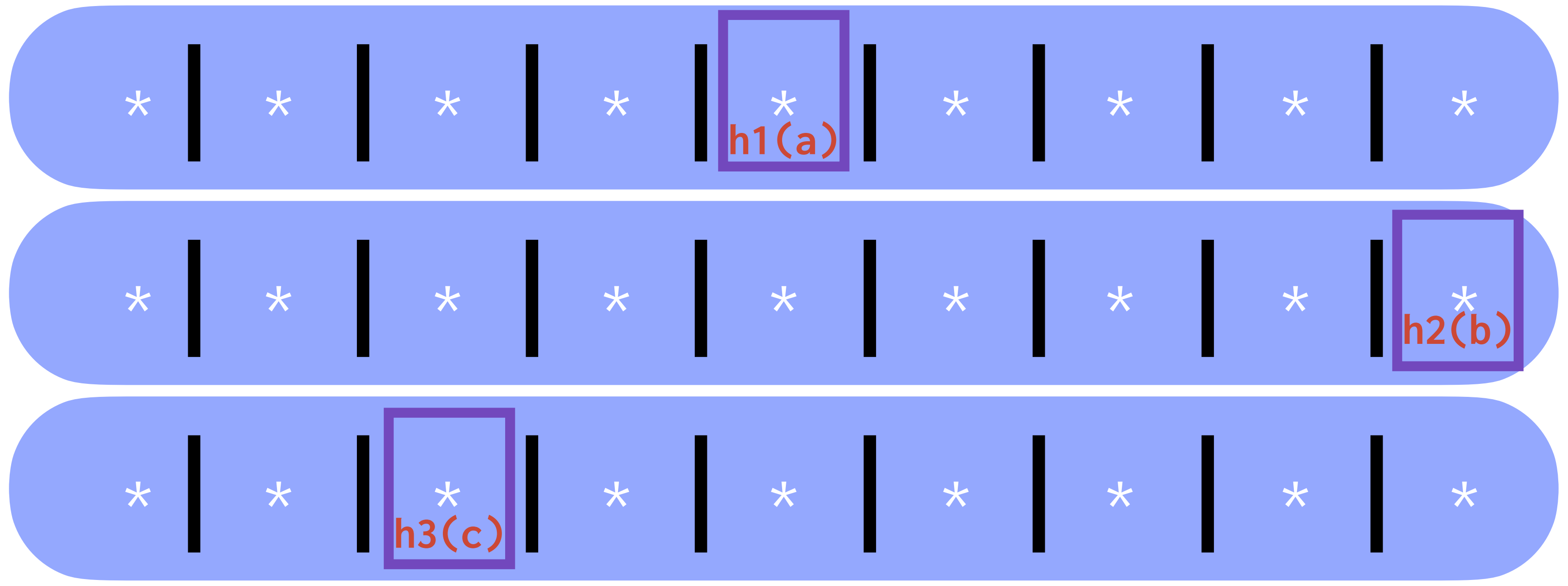
Err: insertions/k

CMS: attack

😈 X

CMS[ ]:
insert
query

Hash



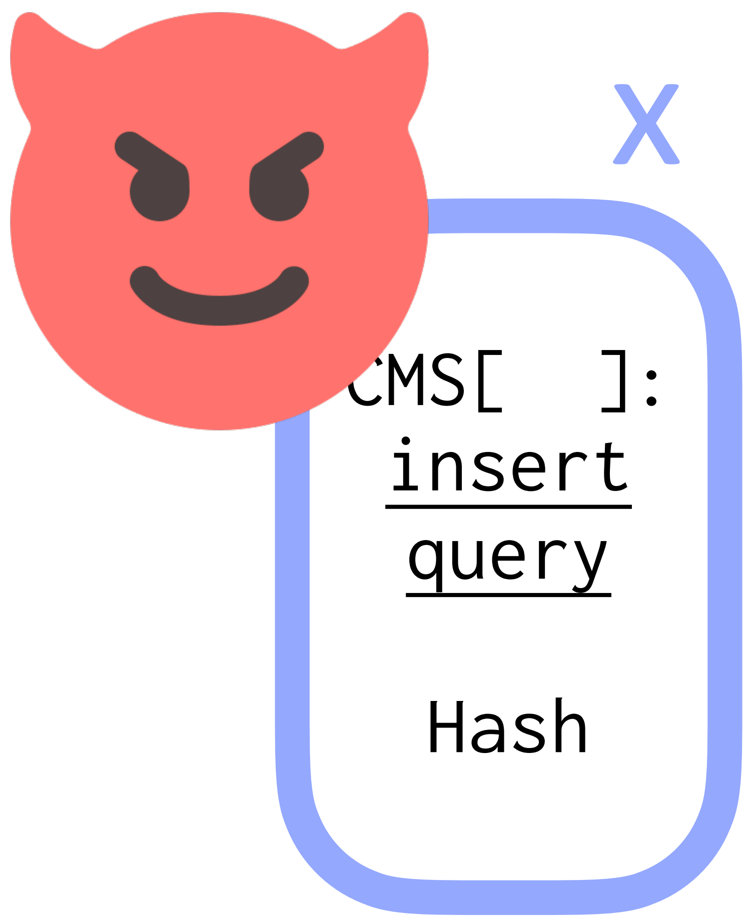
Cover set = {a, b, c}

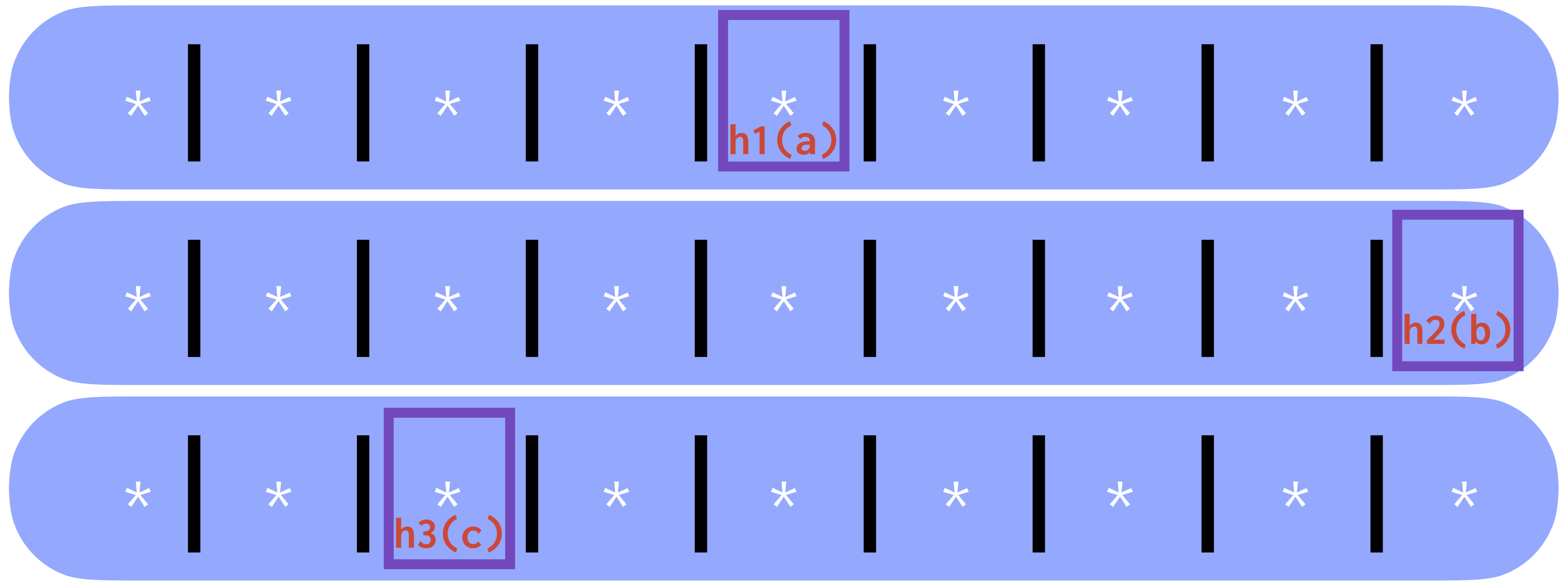
Cover set

2^20

Err: insertions/k

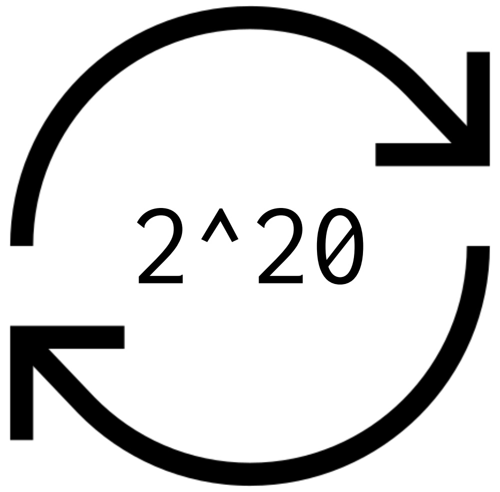
CMS: attack





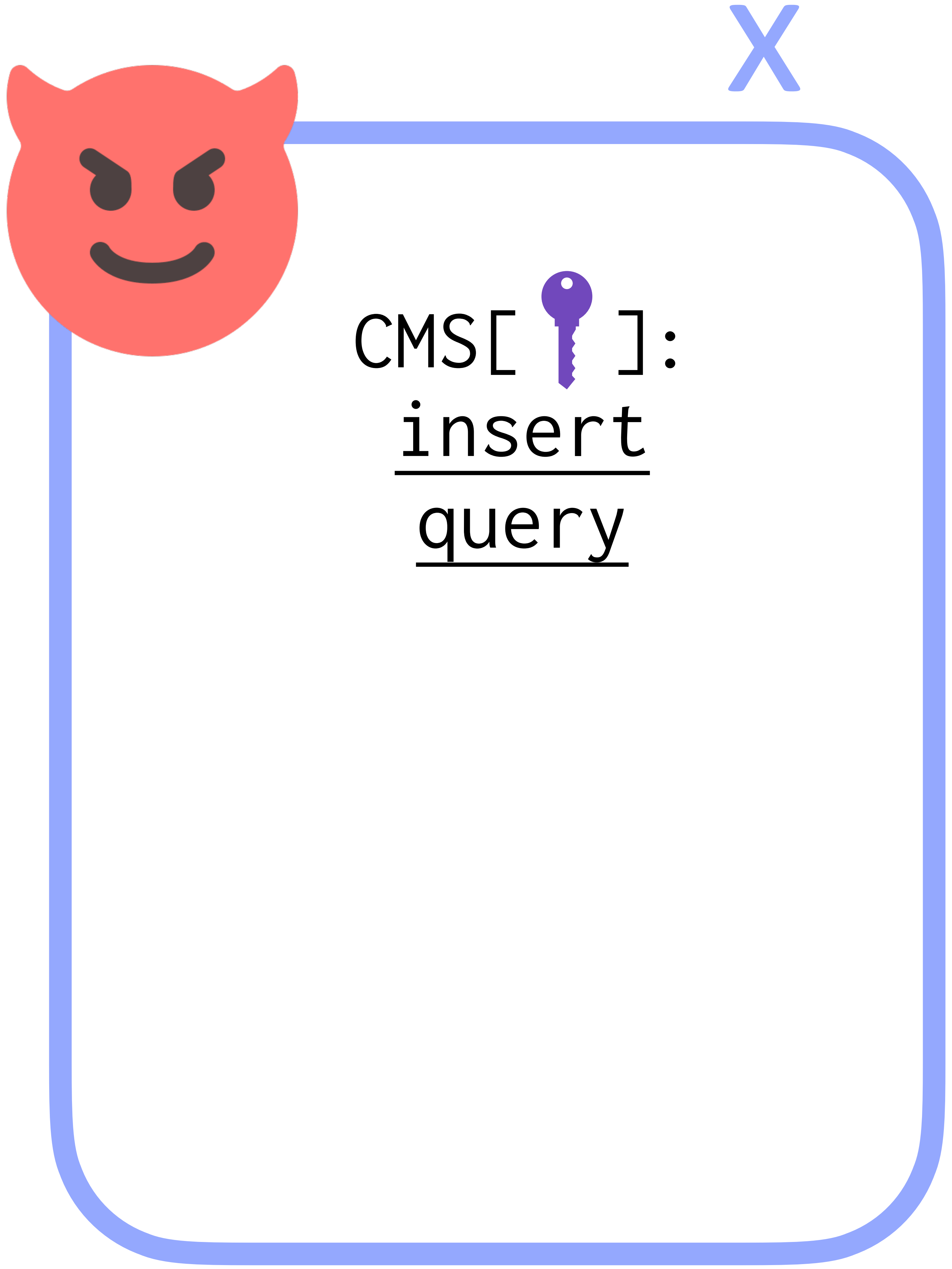
Cover set = {a, b, c}

Cover set

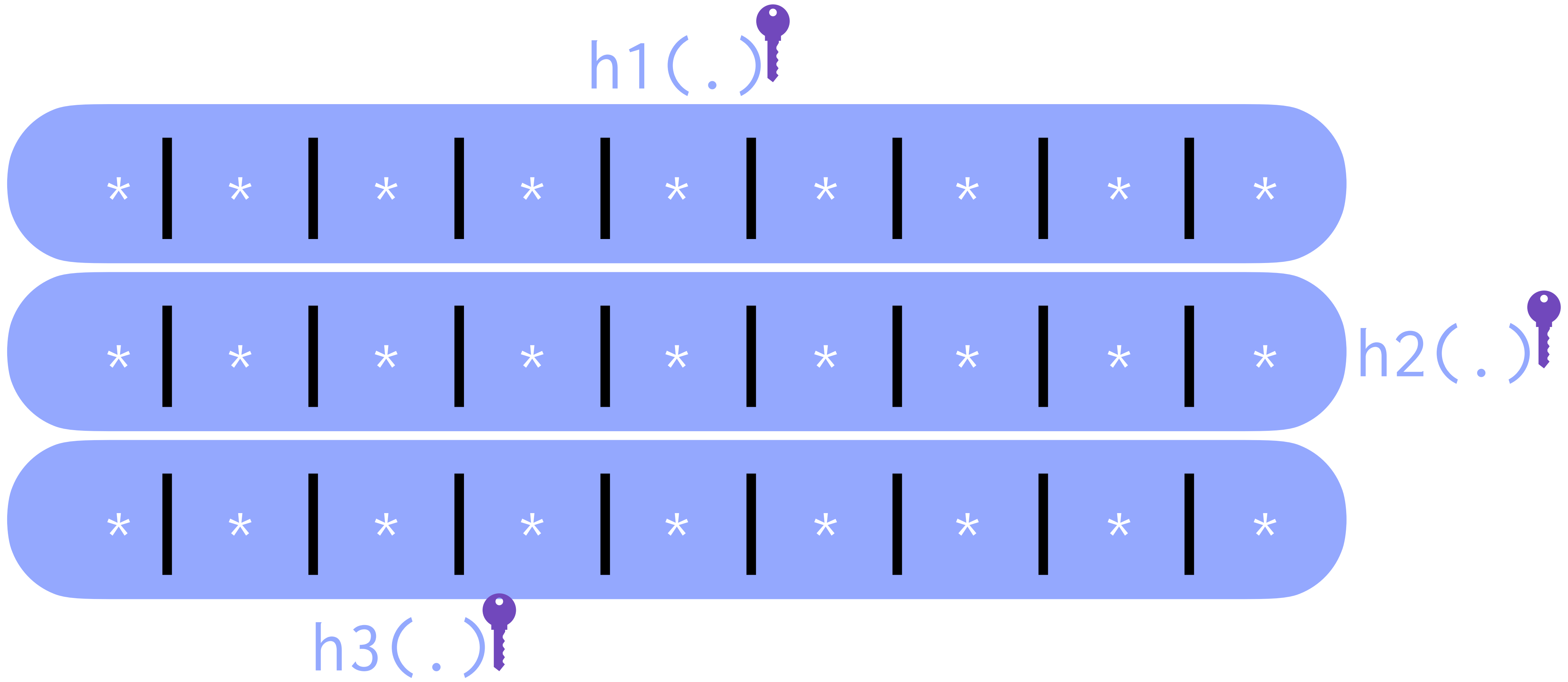


Err: 2^20/3 approx. 350k

CMS: attack model cont.

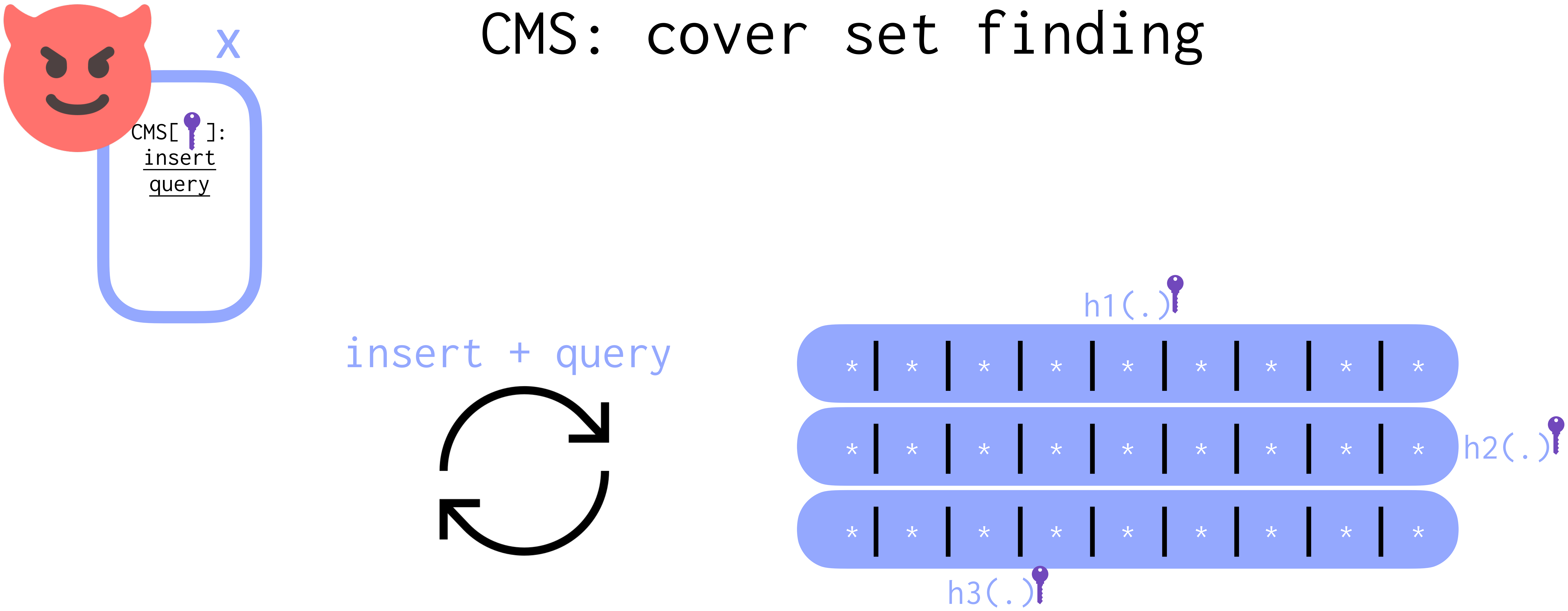


CMS[🔑]:
insert
query

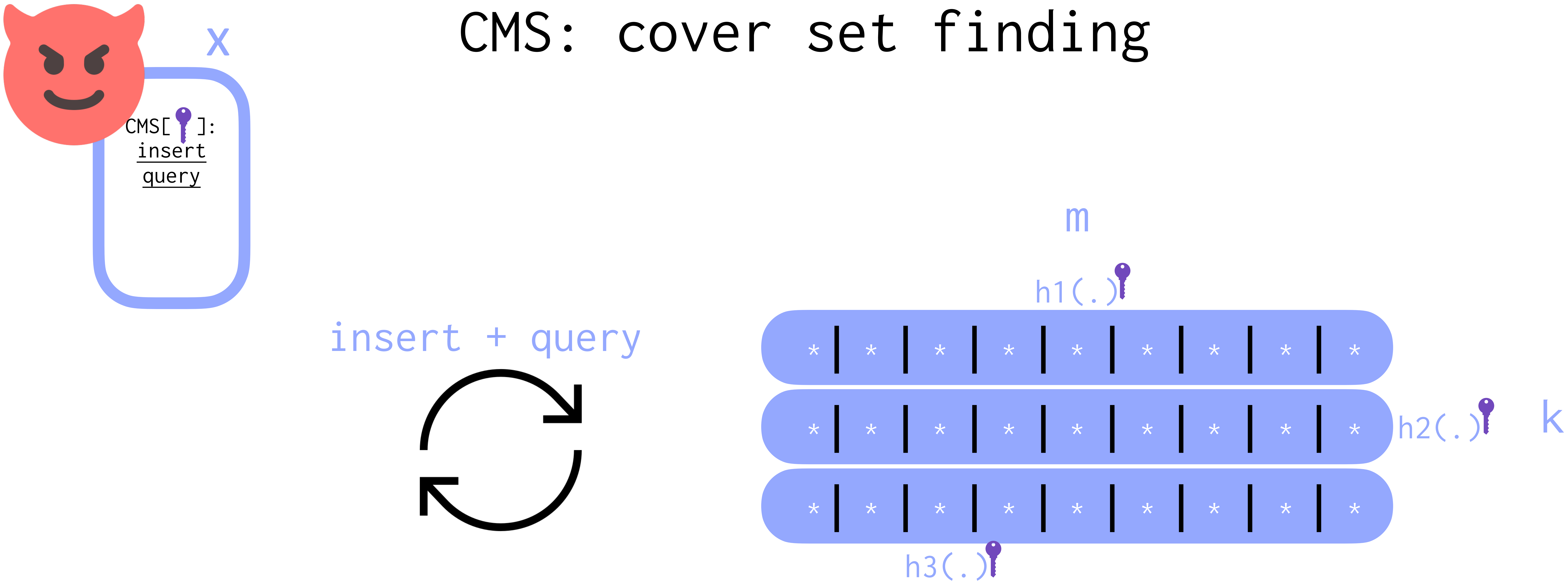


*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

CMS: cover set finding

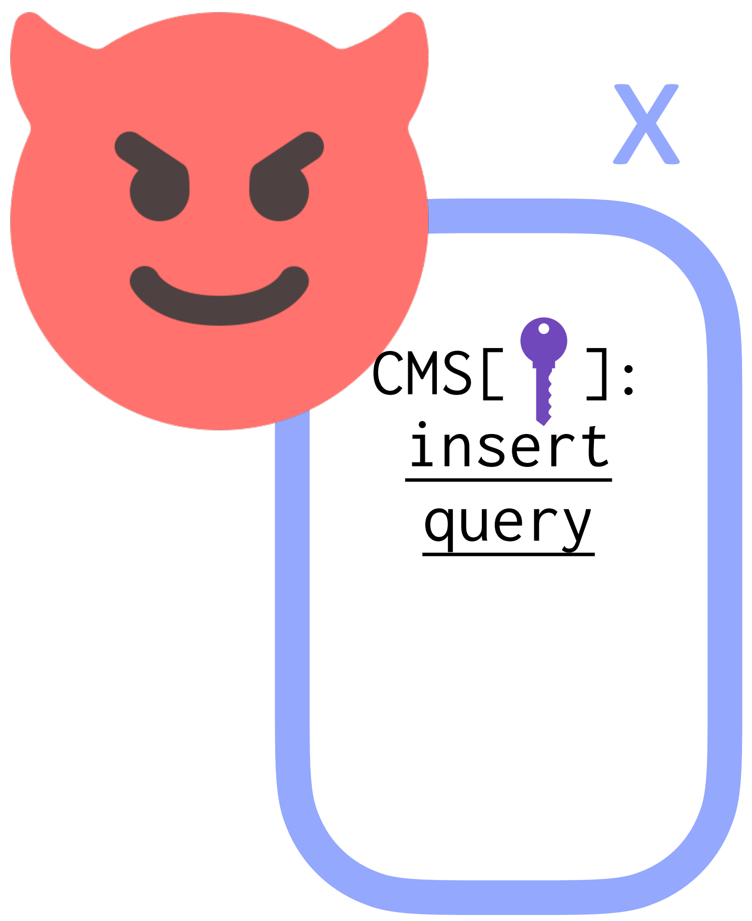


CMS: cover set finding

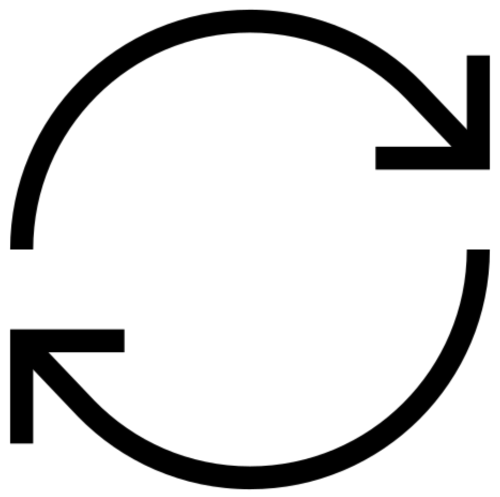


Cover set = {z1, z2, …, zk}

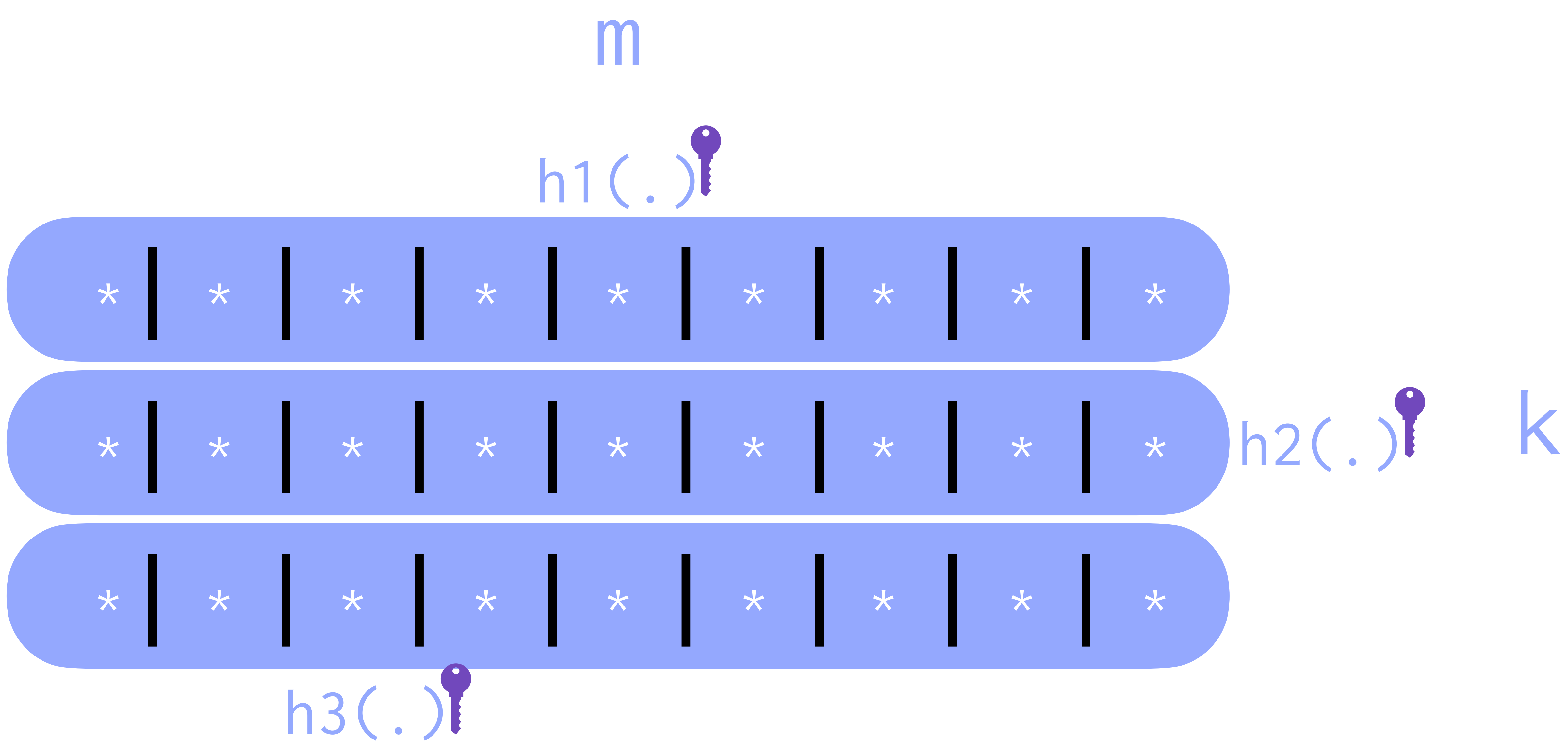
CMS: attack cont.



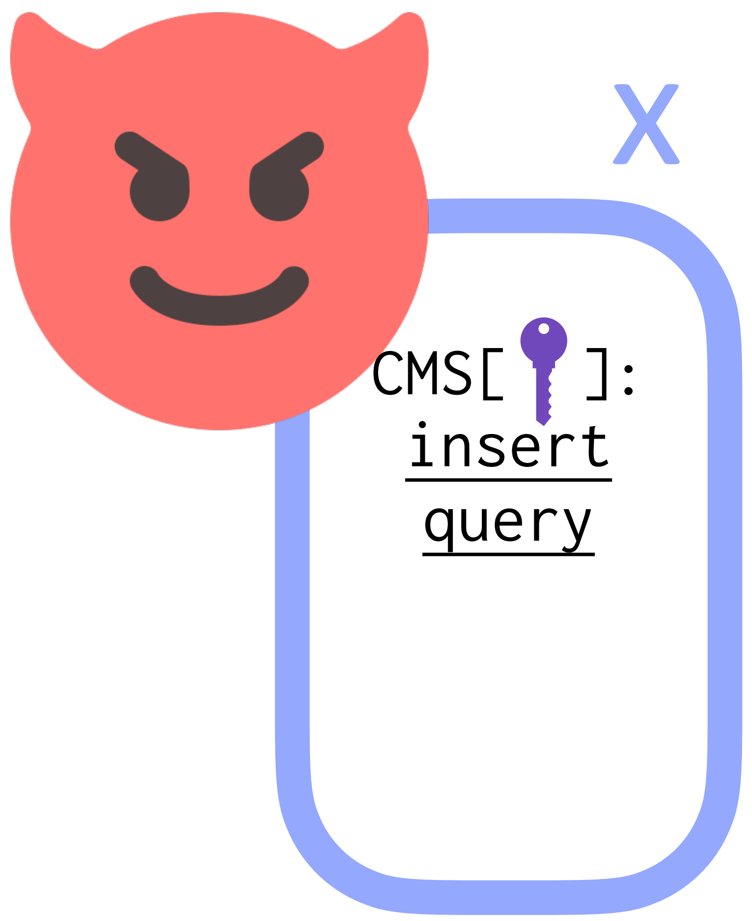
Cover set



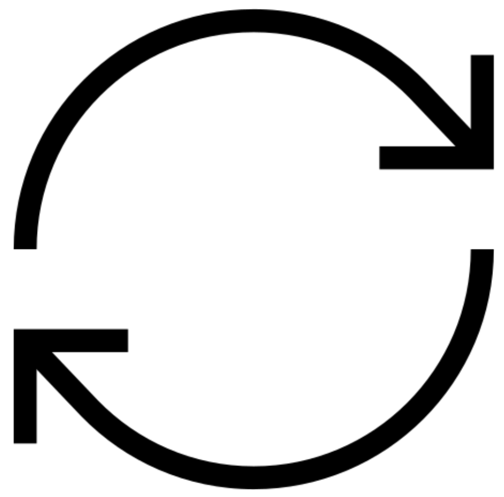
insertions/k-m Hk



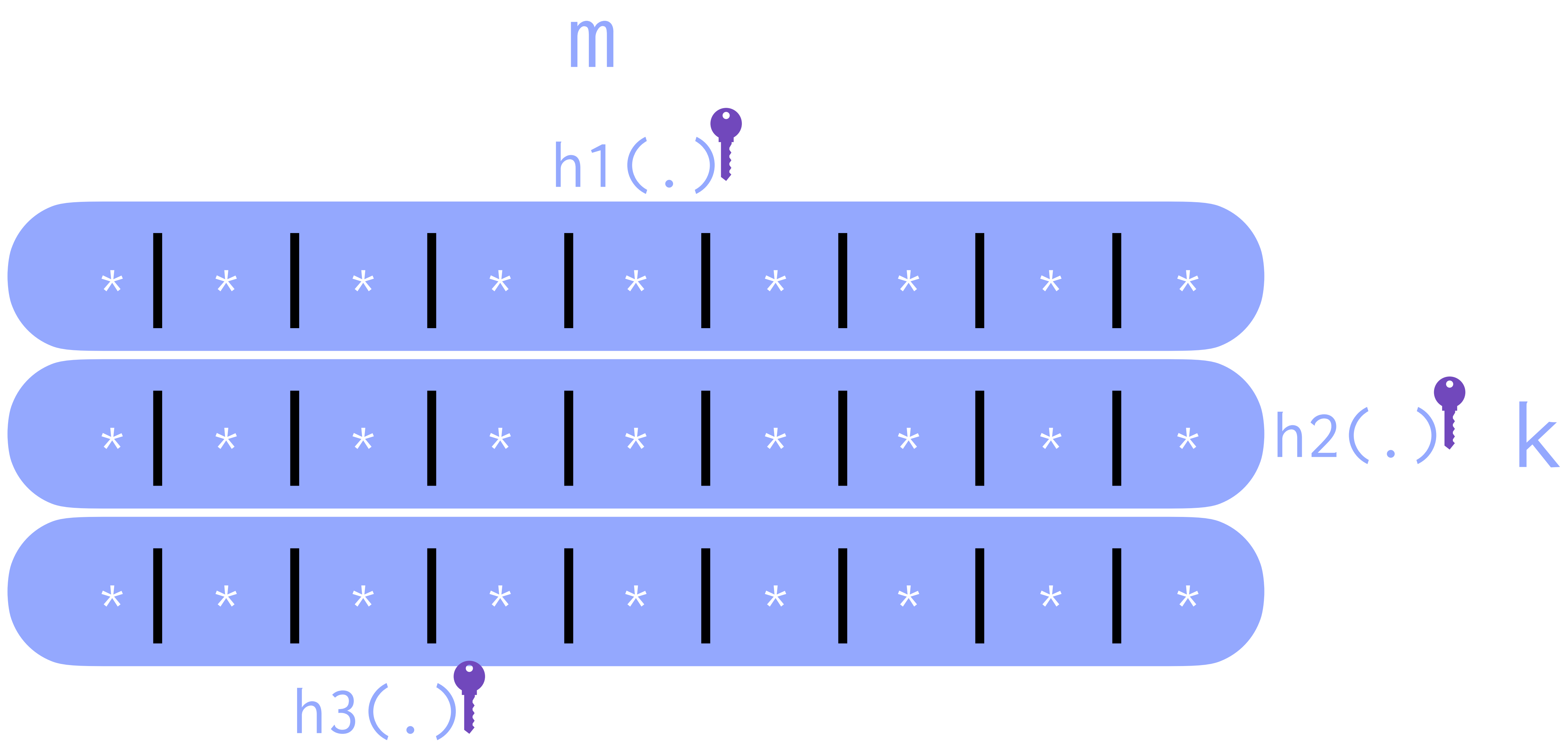
CMS: attack cont.



Cover set



insertions/k-m Hk



Err: insertions/k - m Hk

We have similar attacks against Heavy-Keeper

We have similar attacks against Heavy-Keeper, Count-Sketch, and CMS with conservative updates

Our attacks make

elements **absent** from the stream marked as **heavy**

Our attacks make

elements **absent** from the stream marked as **heavy**
or
high-frequency elements marked as **absent**.

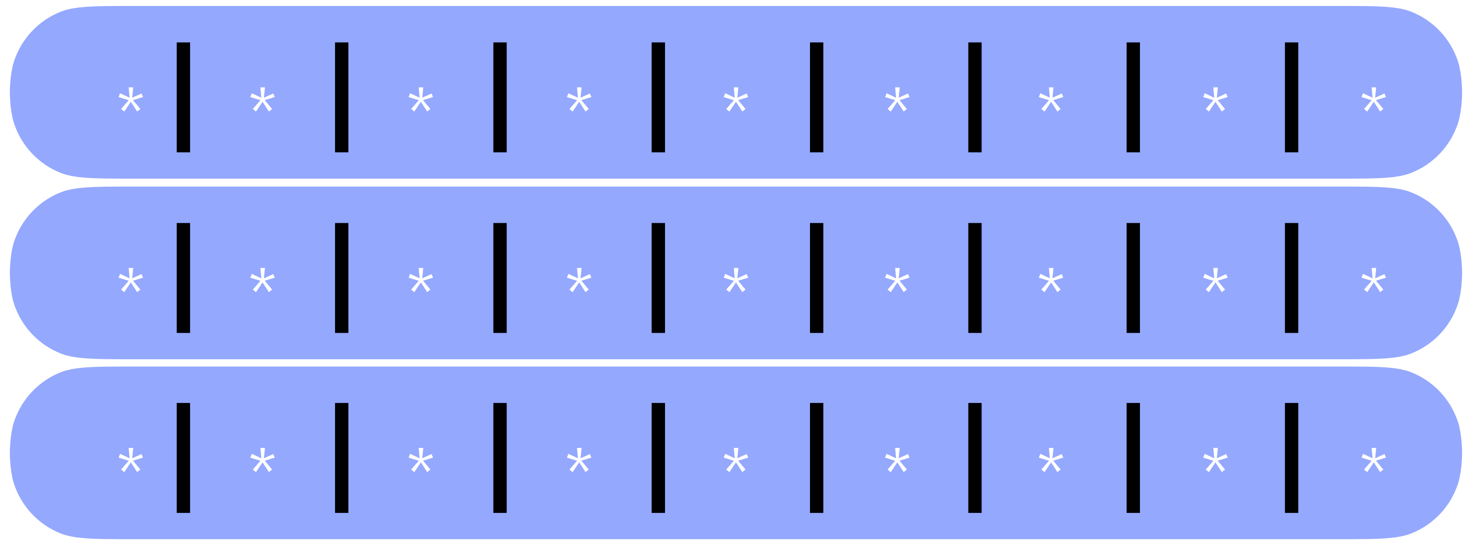
Our attacks make

elements absent from the stream marked as heavy
or
high-frequency elements marked as absent.

Existing CFE are
not adversarially robust

Overestimator + Underestimator

Overestimator + Underestimator

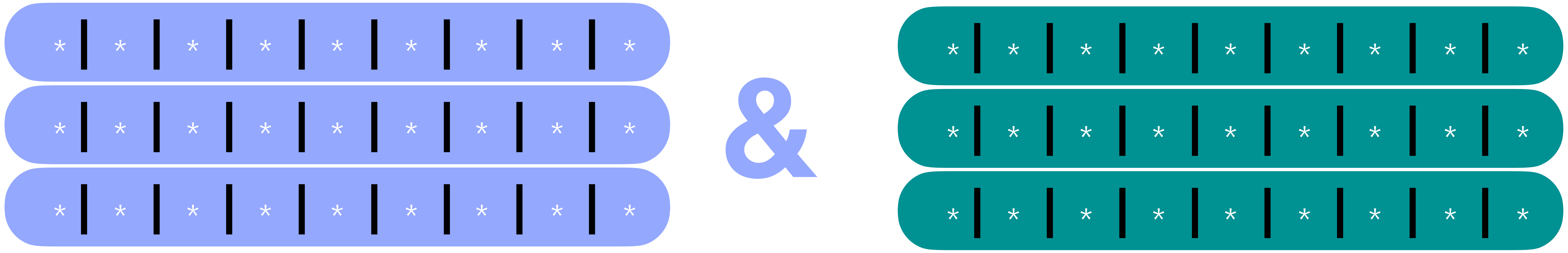


&

CMS M

*HeavyKeeper A

Overestimator + Underestimator

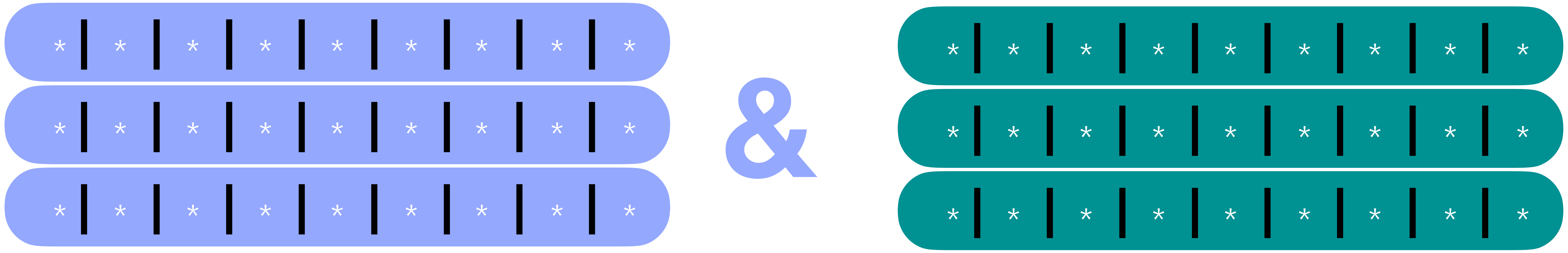


CMS M

*HeavyKeeper A

CMS est & *HeavyKeeper est ---refine---> final est

Count Keeper (CK)

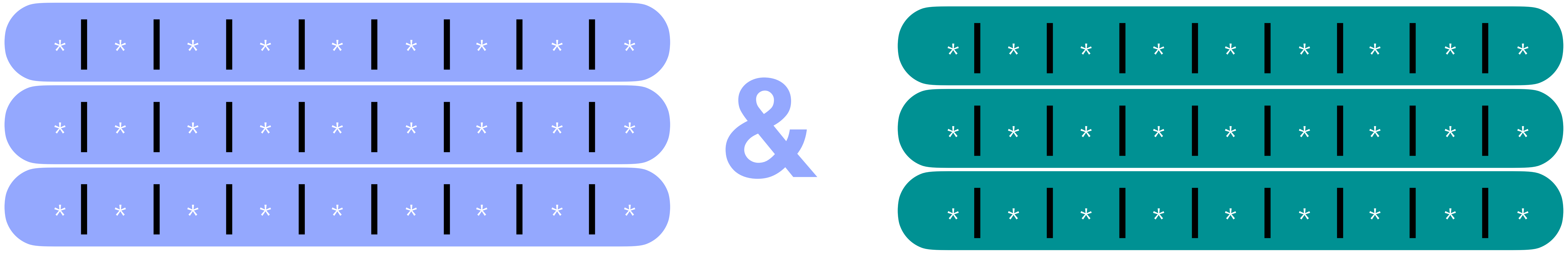


CMS M

*HeavyKeeper A

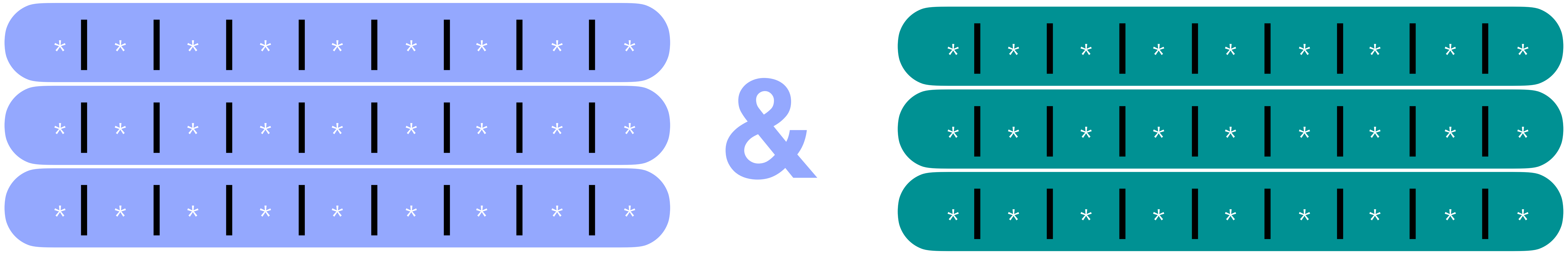
CMS est & *HeavyKeeper est ---refine---> final est

Count Keeper (CK)



CK err < 1/2(CMS est - HK est)

Count Keeper (CK)



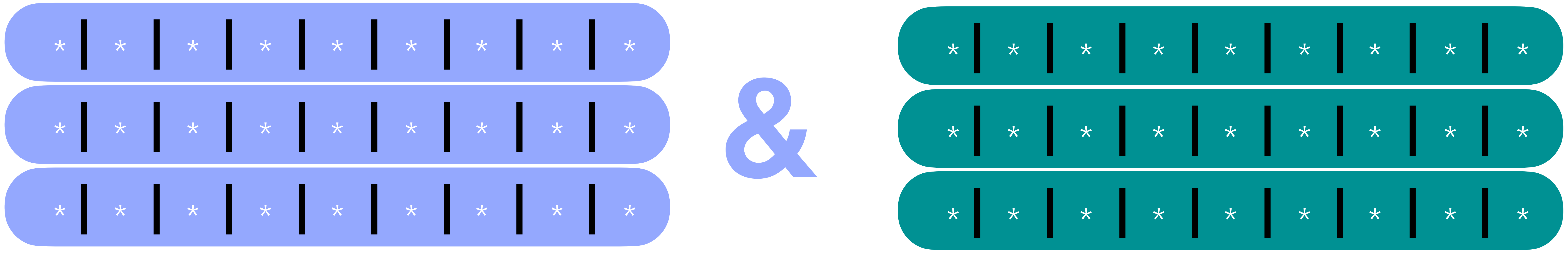
CMS M

*HeavyKeeper A

CK err < 1/2(CMS est - HK est)

+ other error related properties (see our paper) :)

Count Keeper (CK)



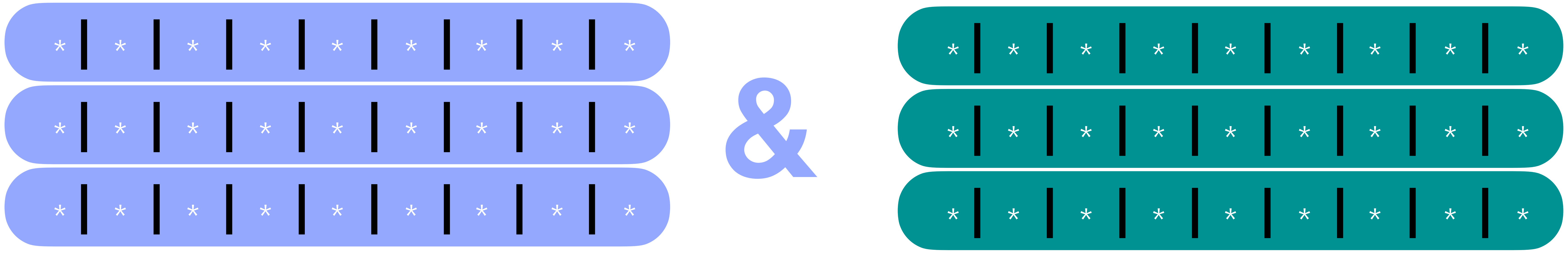
CMS M

*HeavyKeeper A

CK ~~ CMS ~~ HK

Honest setting
experiments

Count Keeper (CK)

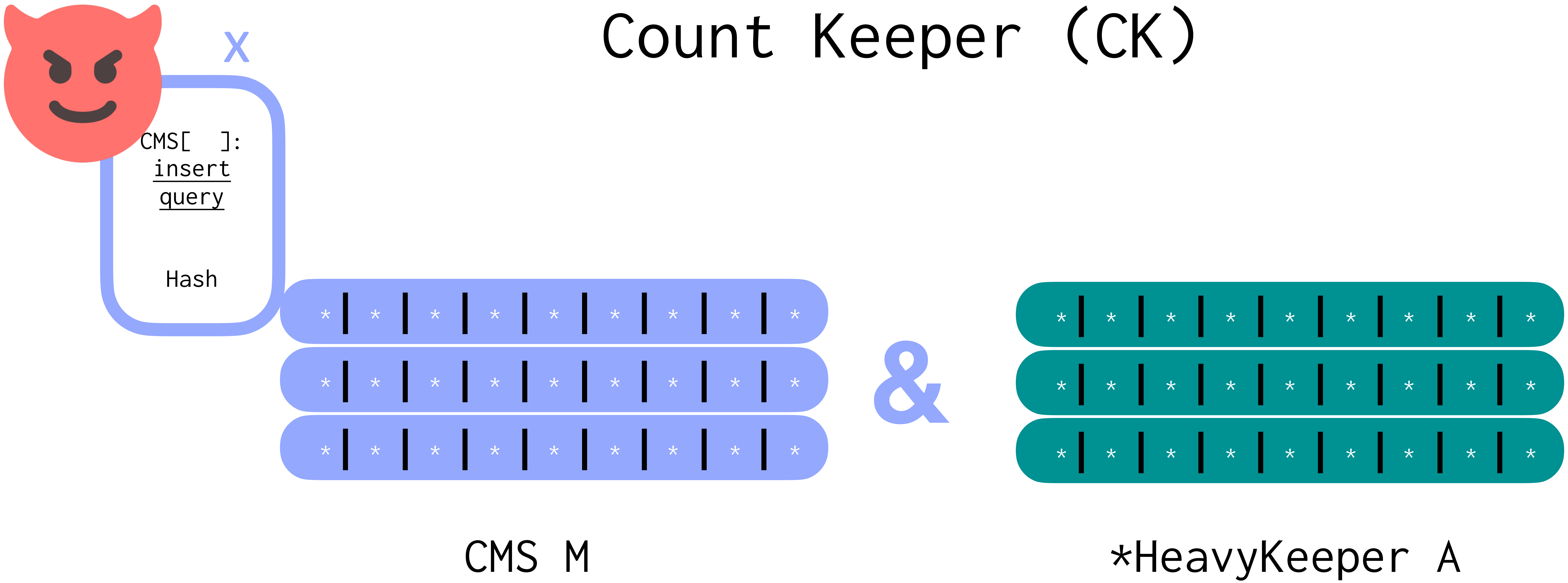


CMS M

*HeavyKeeper A

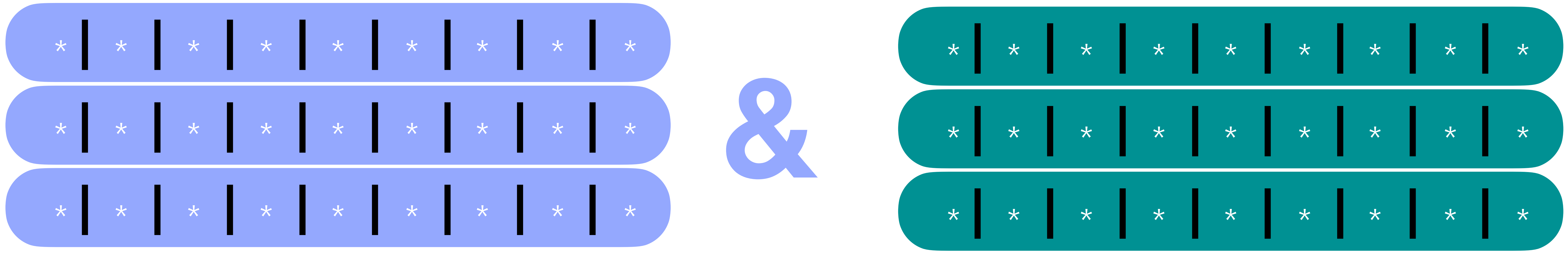
Attacks similar to the CMS ones

Count Keeper (CK)



Err: insertions/(2k)

Count Keeper (CK)



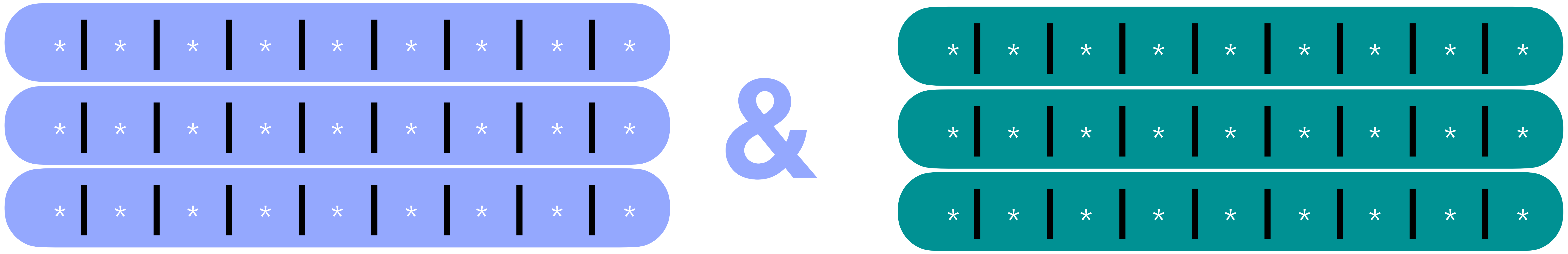
CMS M

*HeavyKeeper A

Err: CK < 1/2 CMS
CK << 1/2 HK

Attack
experiments

Count Keeper (CK)

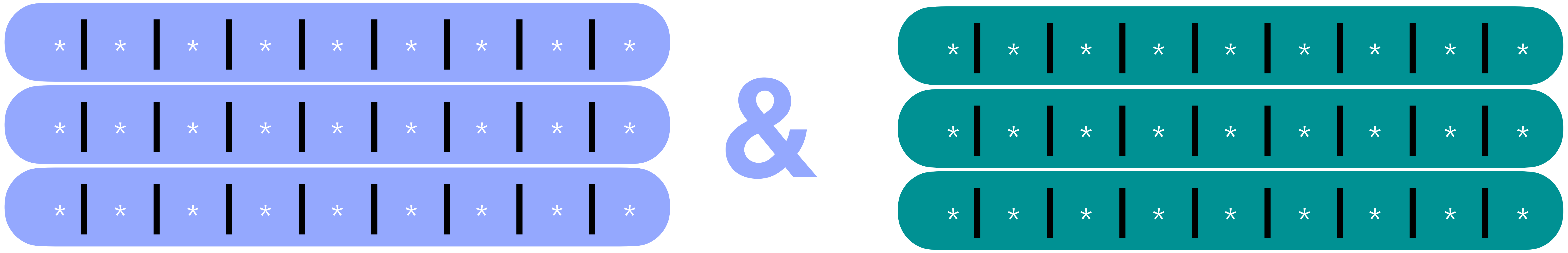


CMS M

*HeavyKeeper A

CK can detect suspicious estimates

Count Keeper (CK)



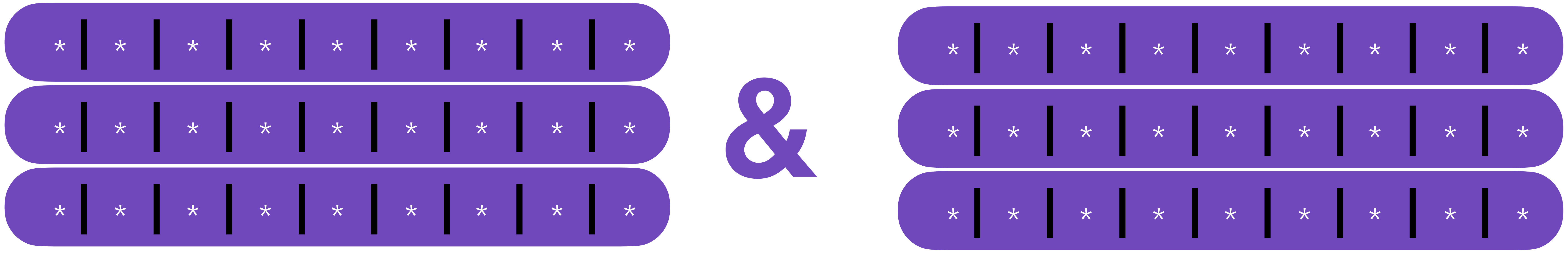
CMS M

&

*HeavyKeeper A

CK can detect suspicious estimates

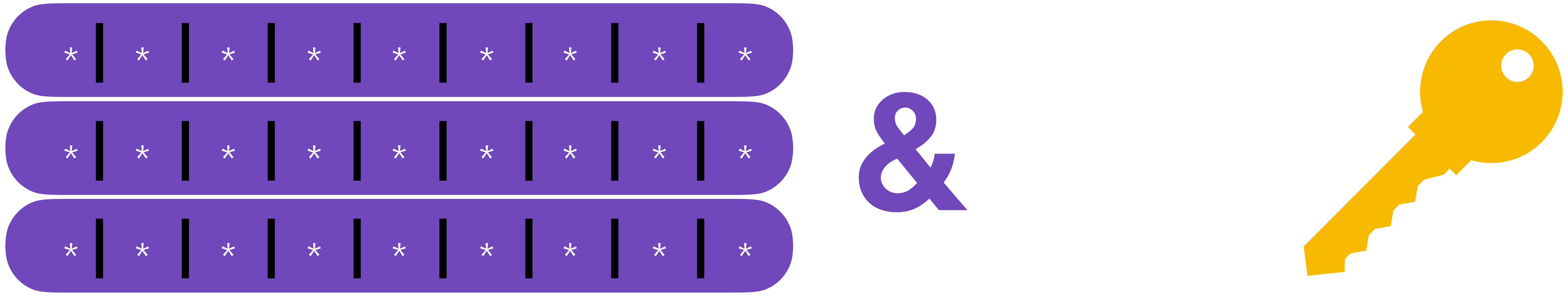
Open problems & Future work



Overestimator ?

Underestimator ?

Open problems & Future work



Thank you!

Paper: https://ia.cr/2023/1366
Code: https://github.com/smarky7CD/cfe-in-adv-envs