Obfuscated Access and Search Patterns in Searchable Encryption

[redacted]

Zhiwei Shang*, Simon Oya*, Andreas Peter*, Florian Kerschbaum*

University of Waterloo

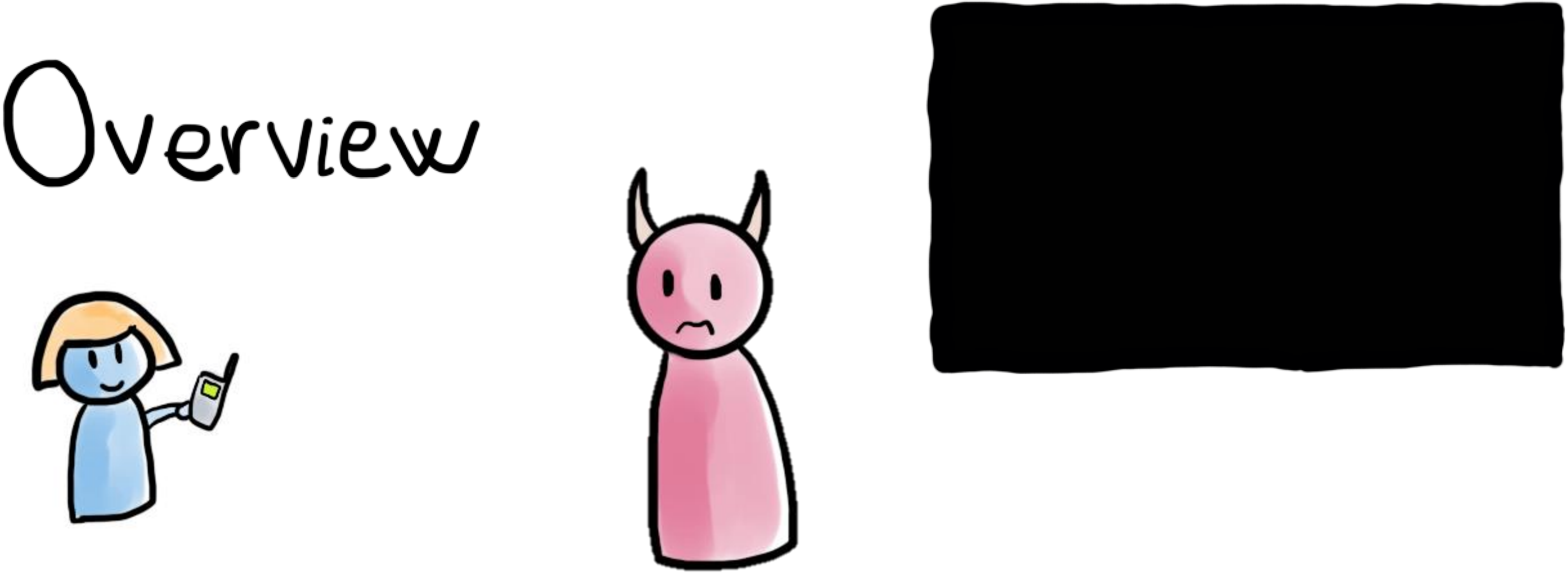
University of Twente

NDSS'21

Overview

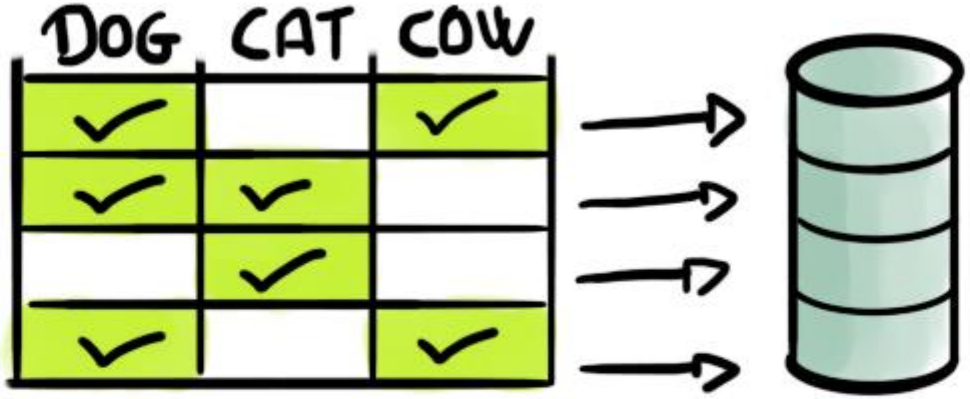
[redacted]

Overview



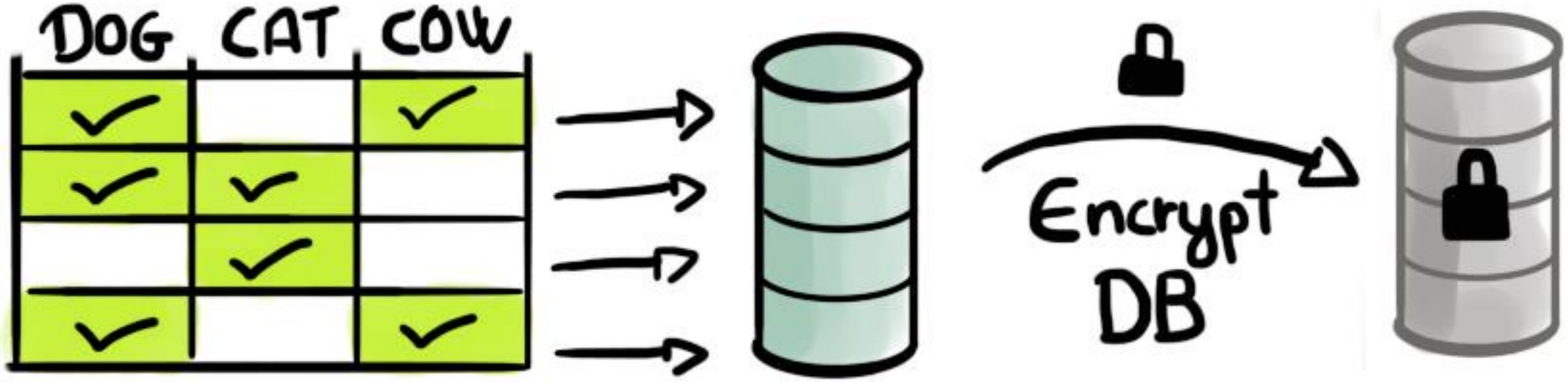
Overview

[redacted]



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

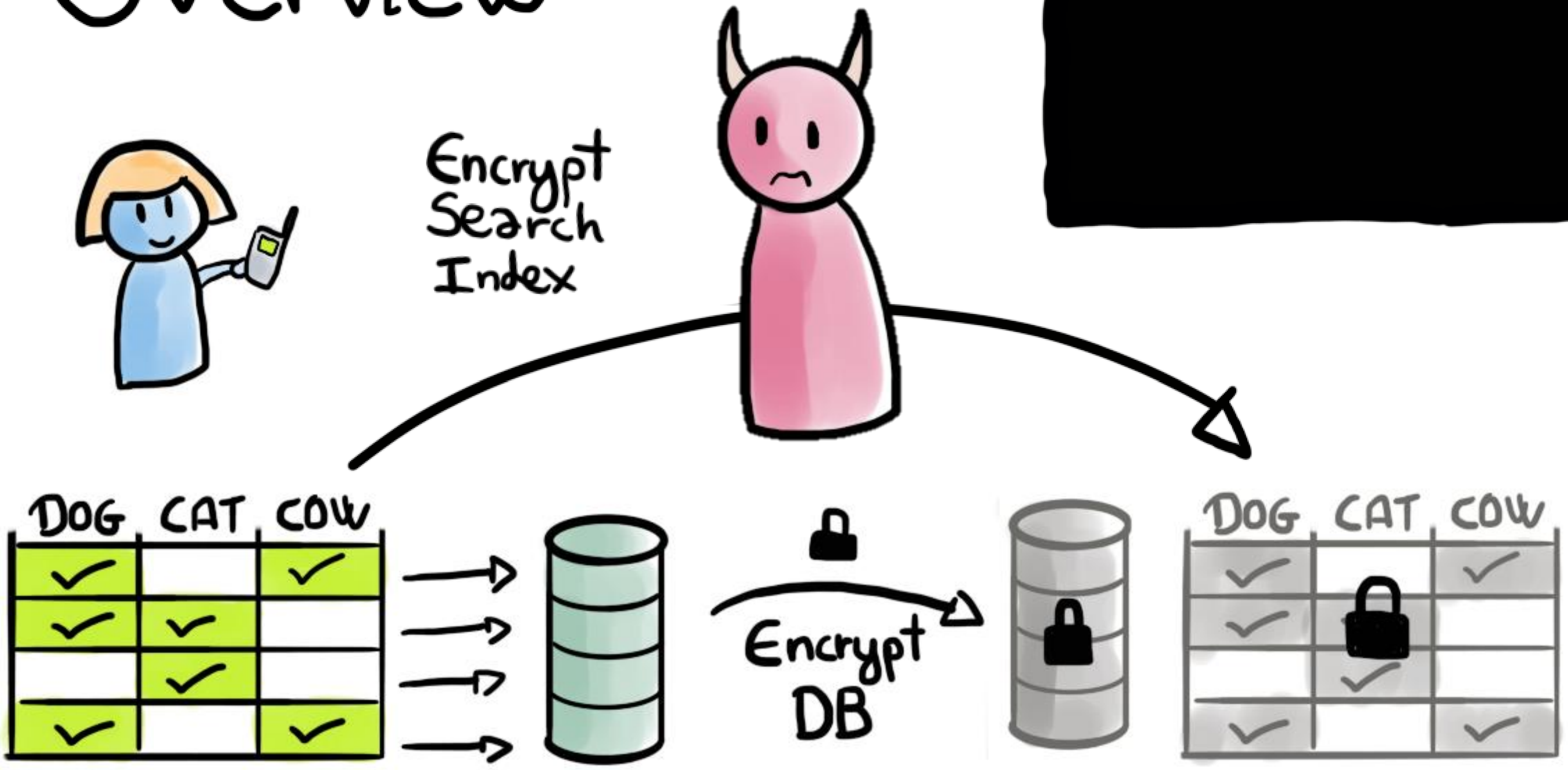
Overview



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

Overview

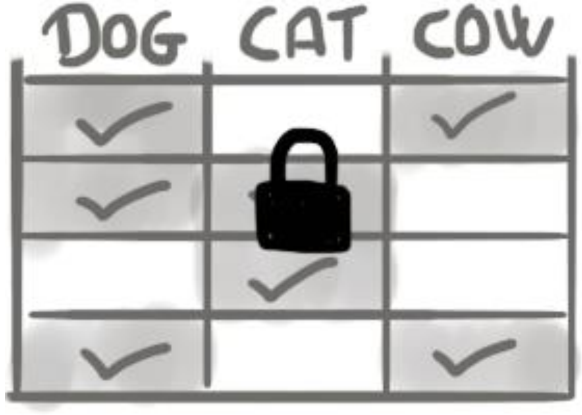
[redacted]



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

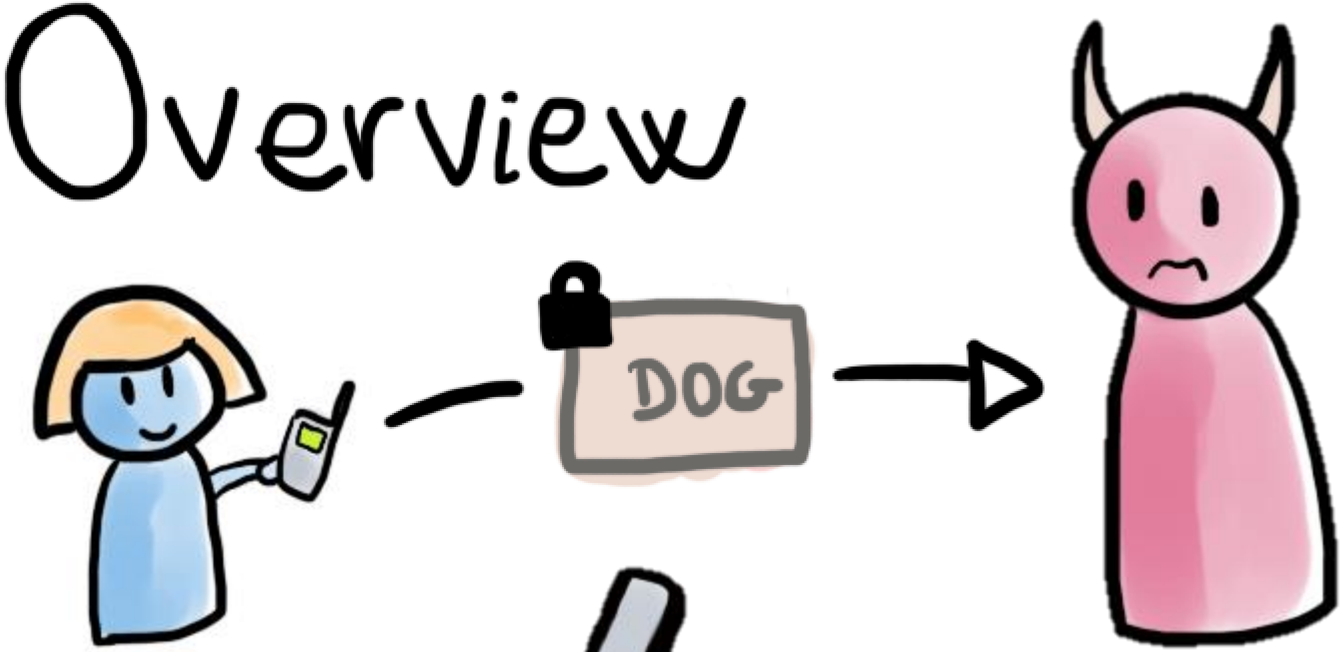
Overview

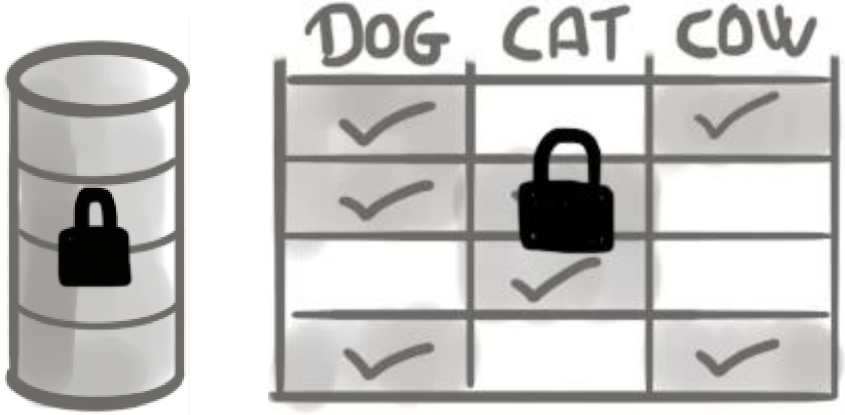
[redacted]



DOG	CAT	COW
✓		✓
✓	🔒	
	✓	
✓		✓

Overview

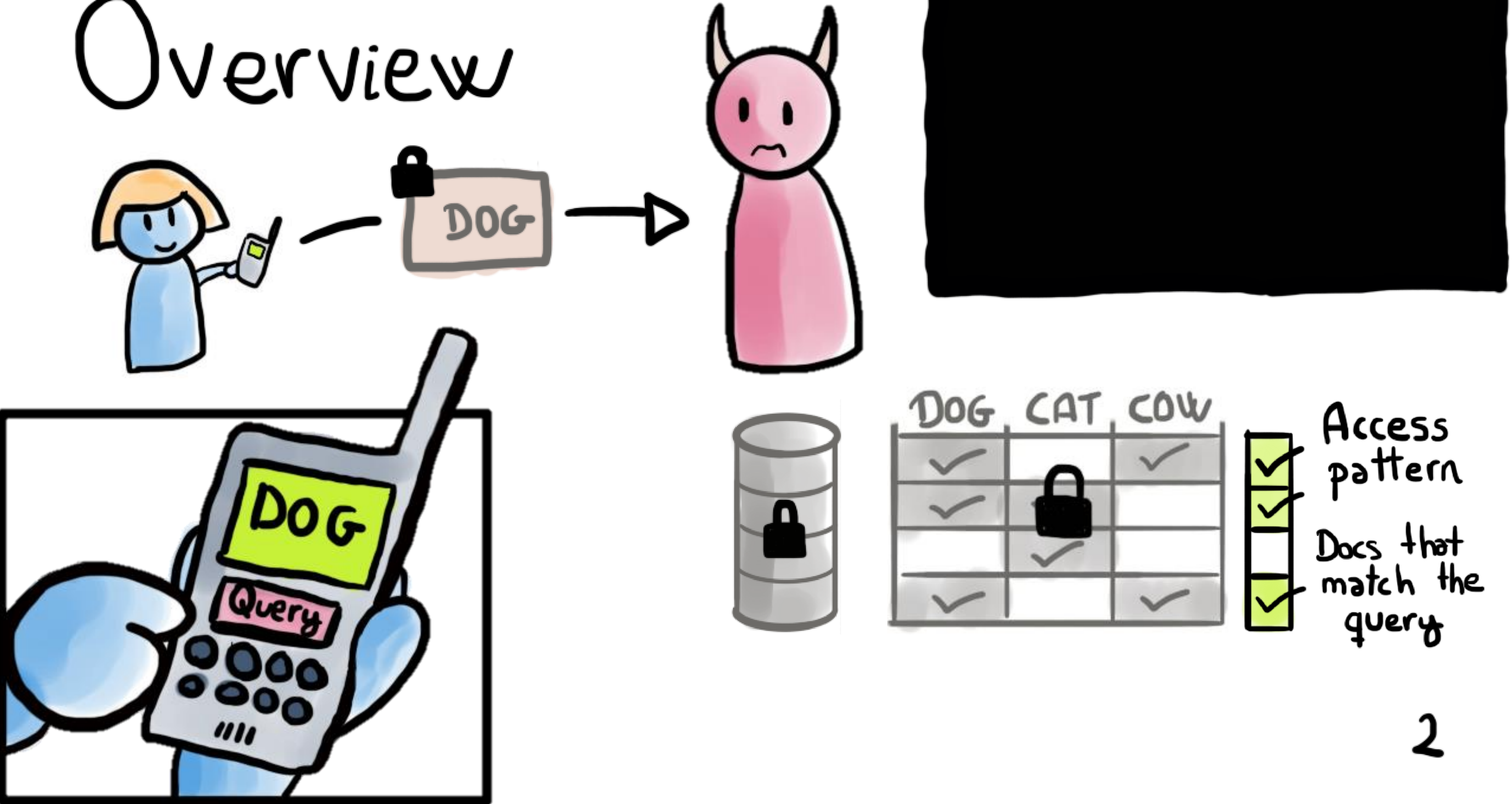




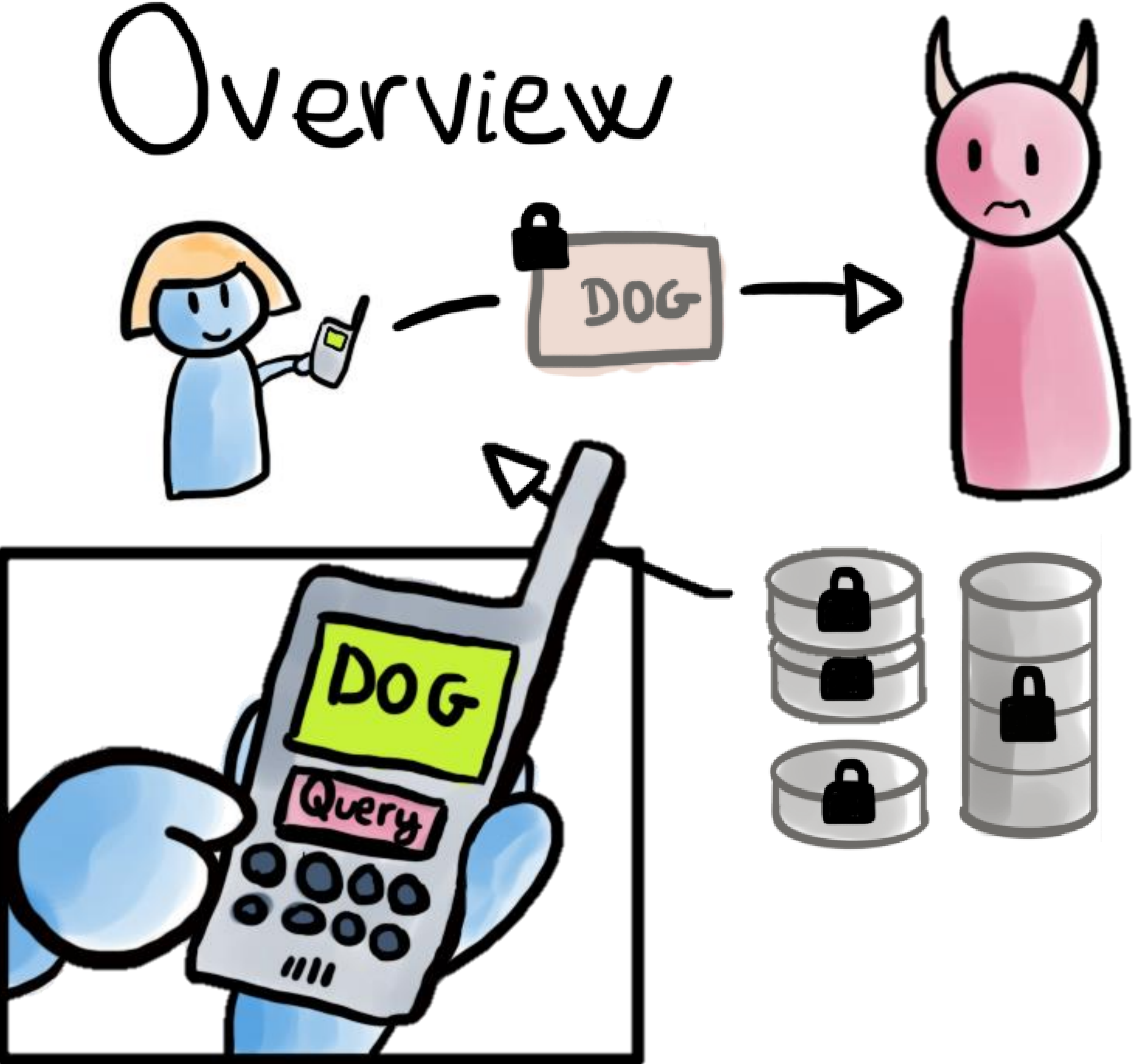
DOG	CAT	COW
✓		✓
✓		
	✓	
✓		✓

Overview

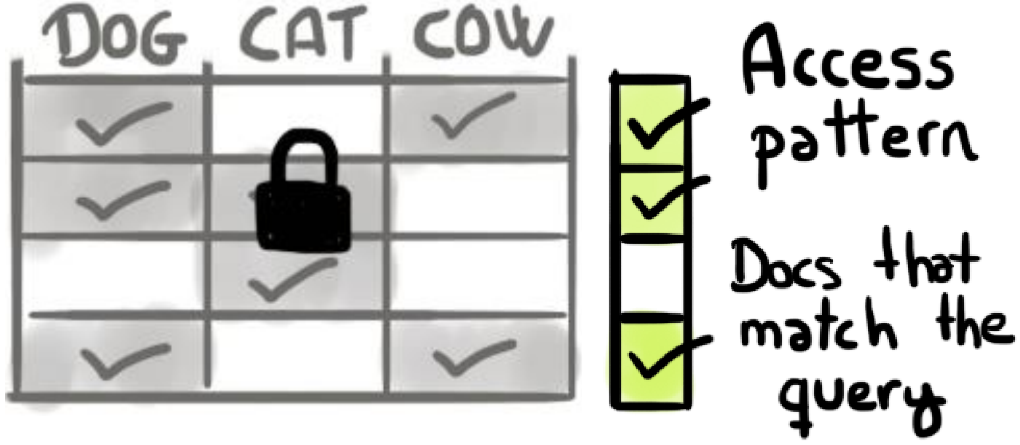
[redacted]



Overview

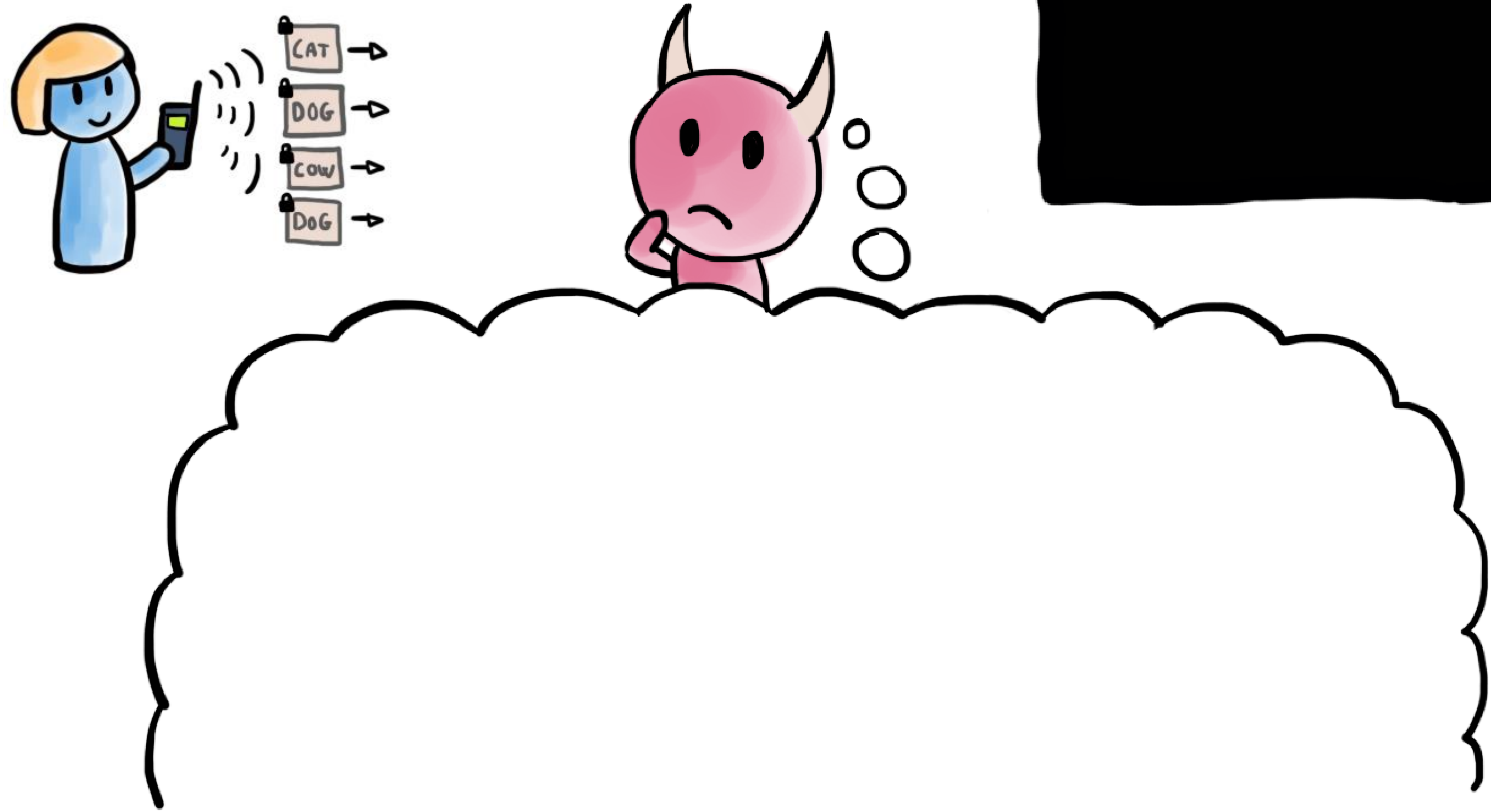


[redacted]

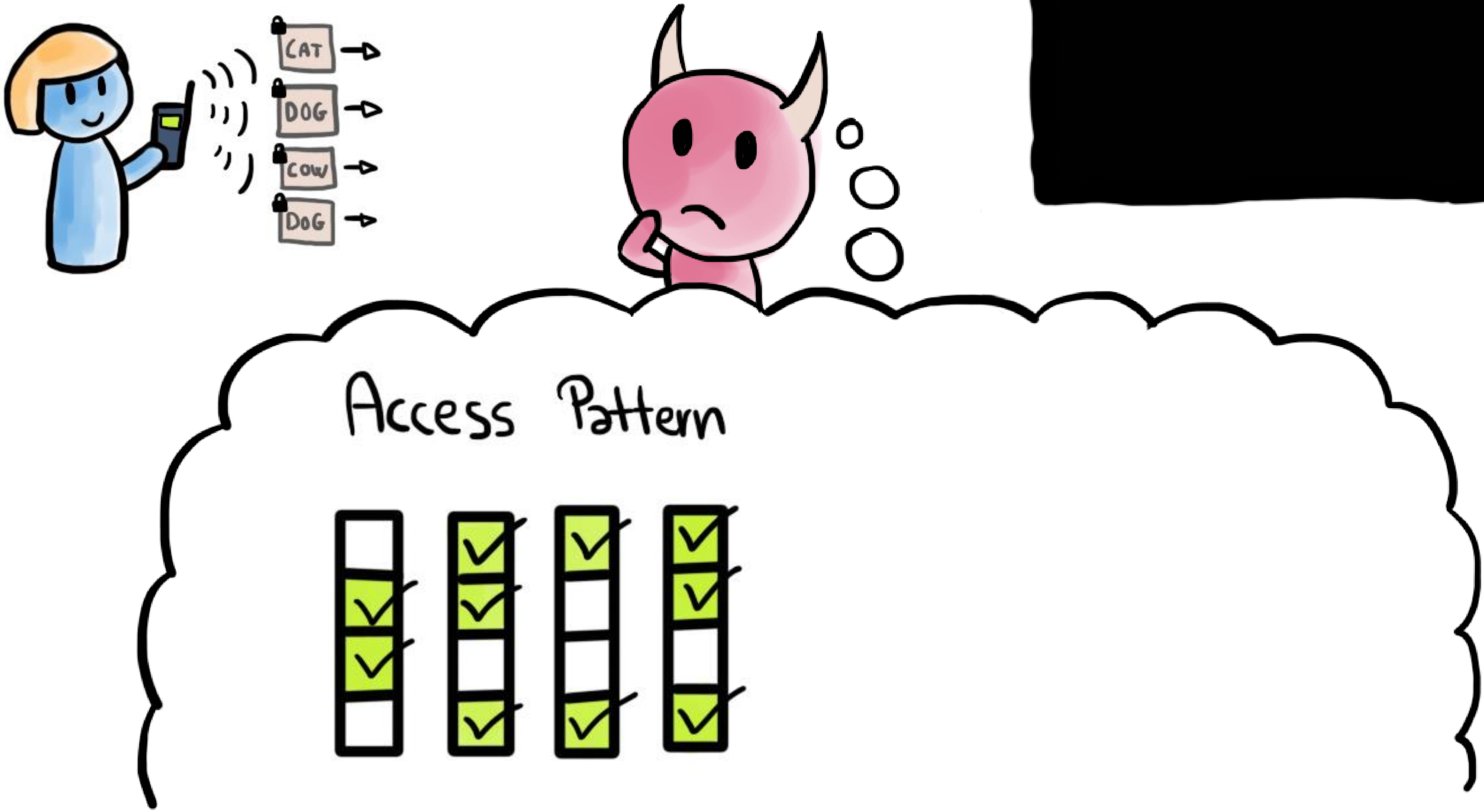


DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

Overview

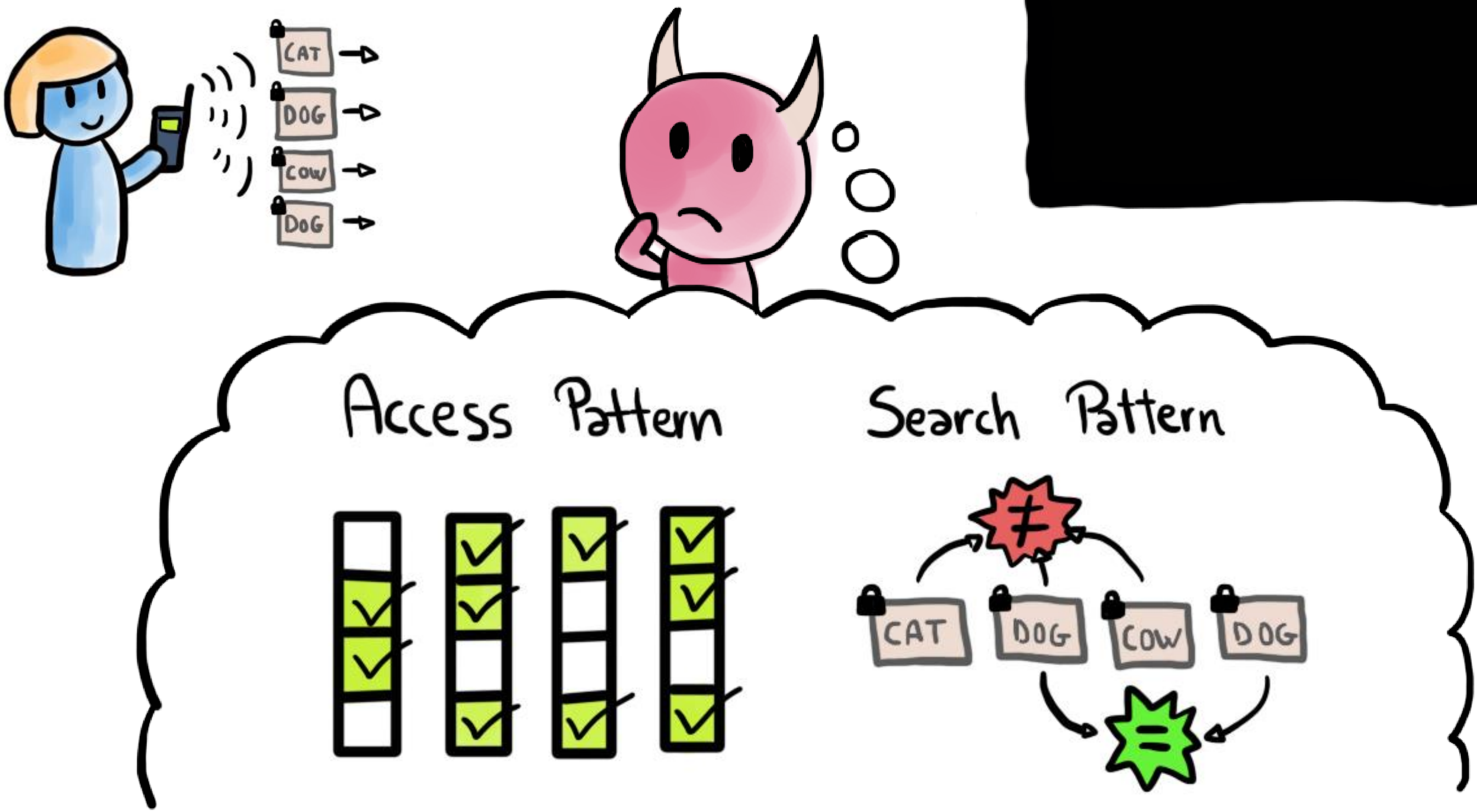


Overview



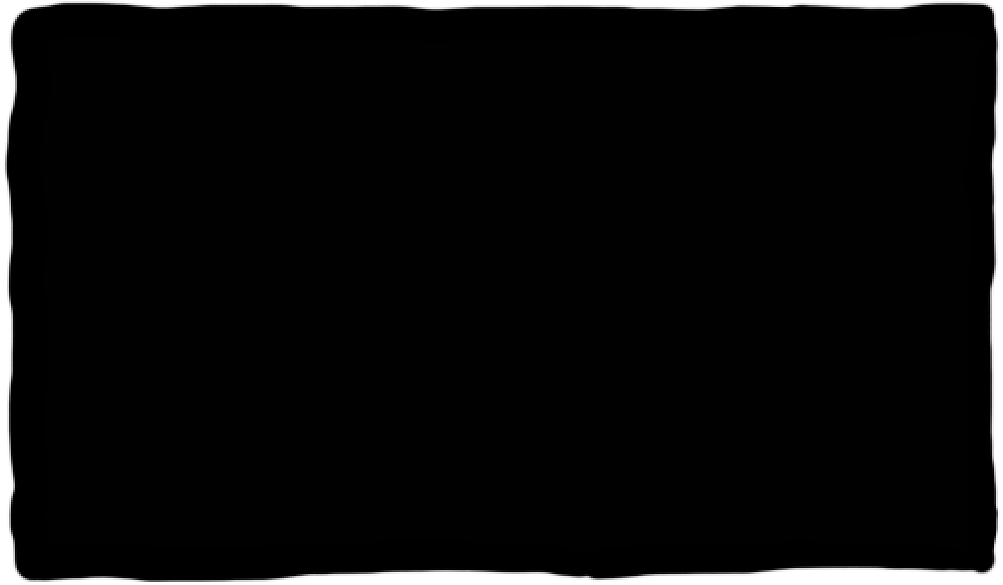
Overview

[redacted]



Hiding Access Pattern

DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓



Hiding Access Pattern

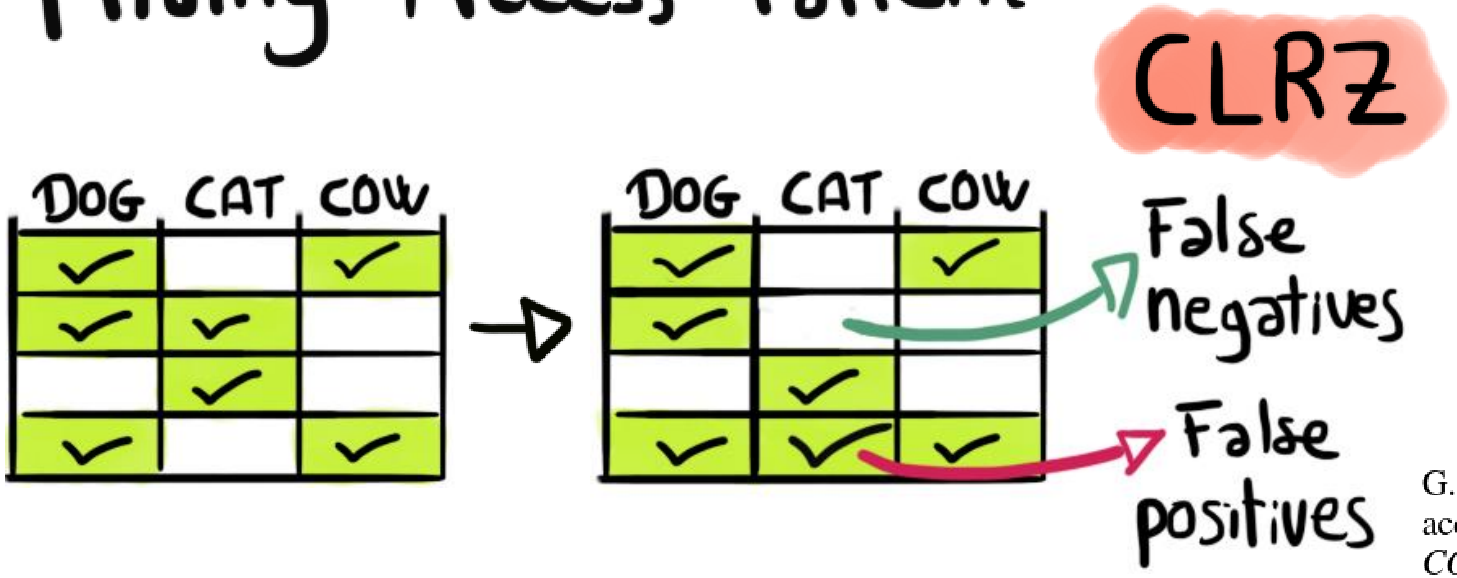
DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

CLRZ

[redacted]

G. Chen, T.-H. Lai, M. K. Reiter, and Y. Zhang, "Differentially private access patterns for searchable symmetric encryption," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 810–818.

Hiding Access Pattern



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

→

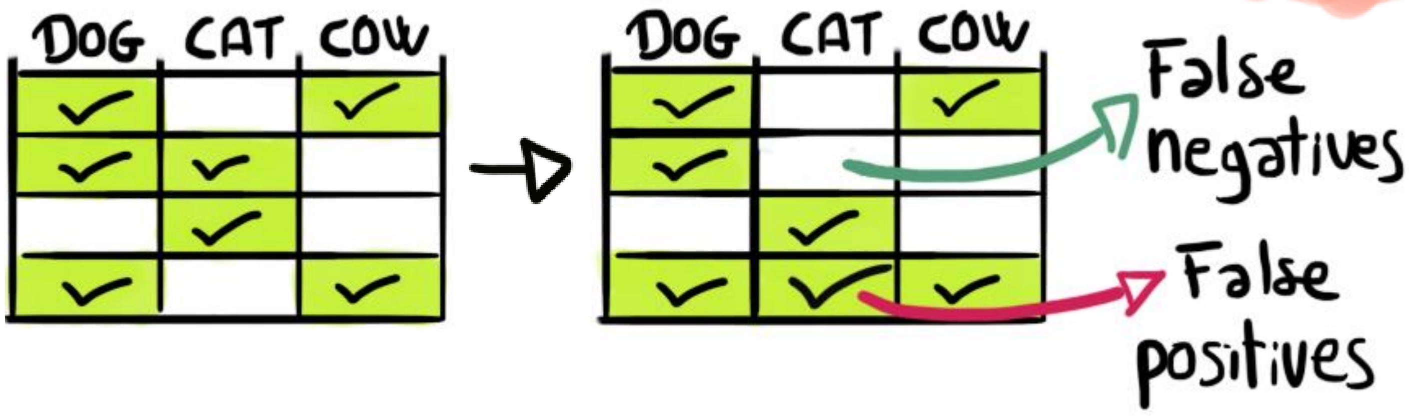
DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

[redacted]

G. Chen, T.-H. Lai, M. K. Reiter, and Y. Zhang, "Differentially private access patterns for searchable symmetric encryption," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 810–818.

Hiding Access Pattern

CLRZ



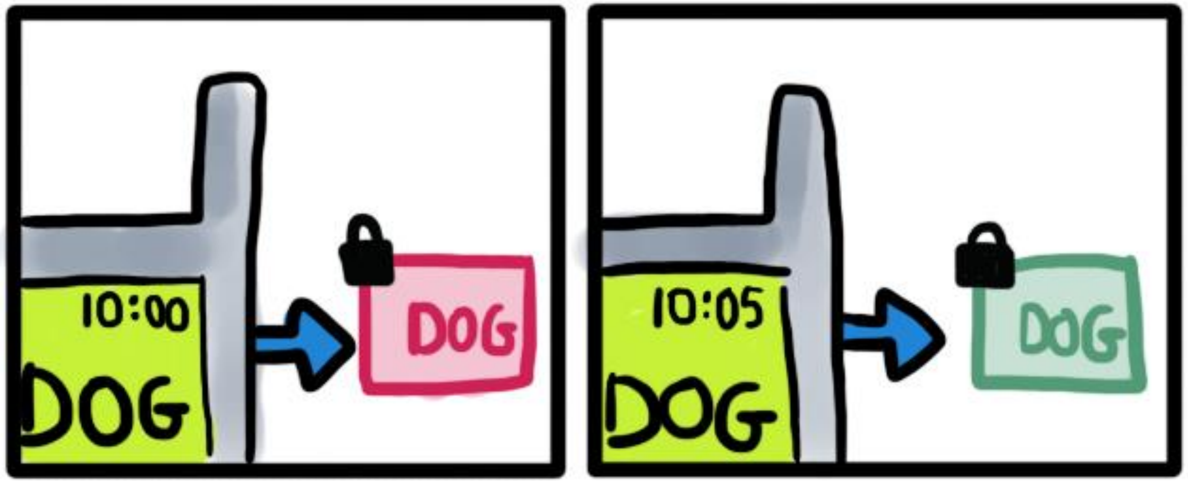
DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

→

DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

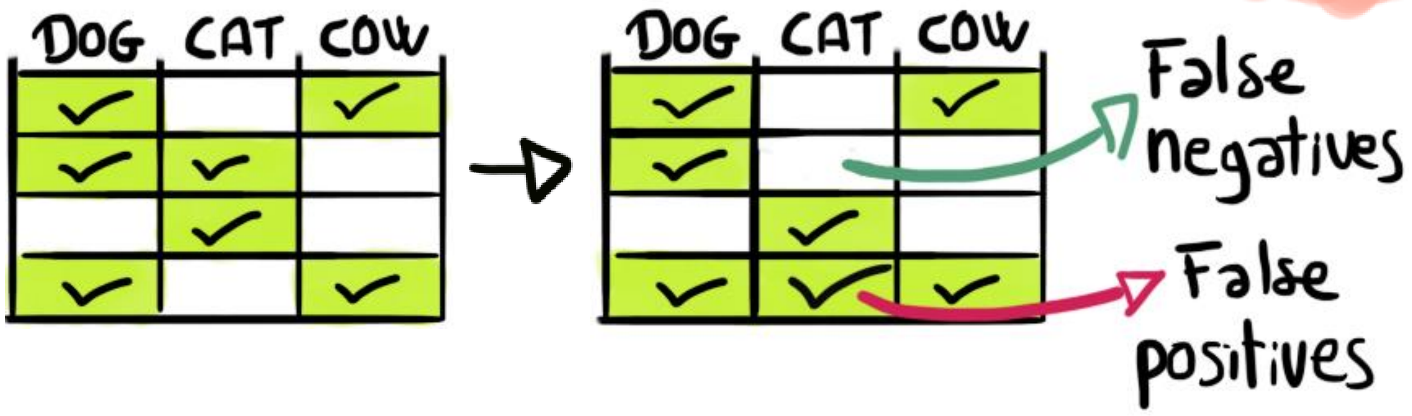
G. Chen, T.-H. Lai, M. K. Reiter, and Y. Zhang, "Differentially private access patterns for searchable symmetric encryption," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 810–818.

Hiding Search Pattern?



Hiding Access Pattern

CLRZ



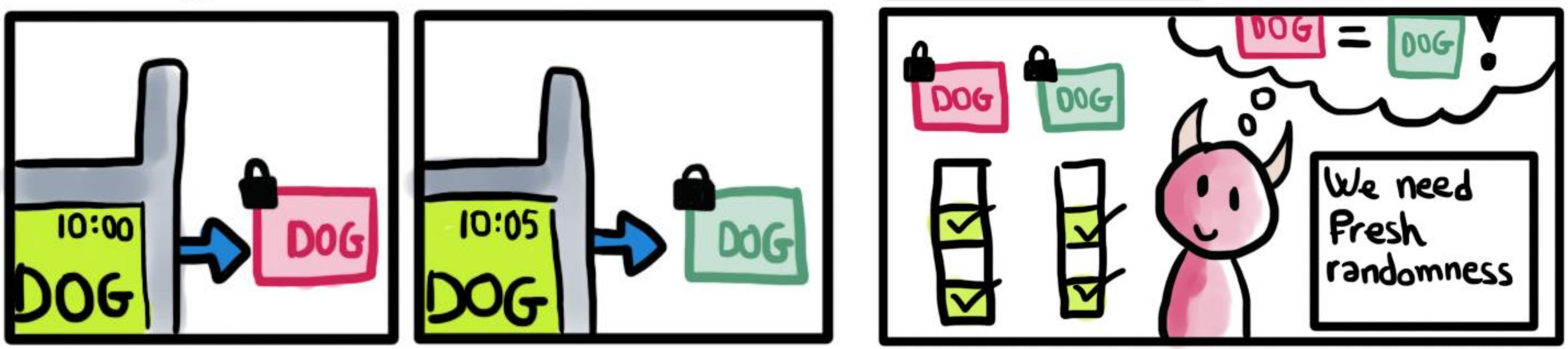
DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

→

DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

G. Chen, T.-H. Lai, M. K. Reiter, and Y. Zhang, "Differentially private access patterns for searchable symmetric encryption," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 810–818.

Hiding Search Pattern?

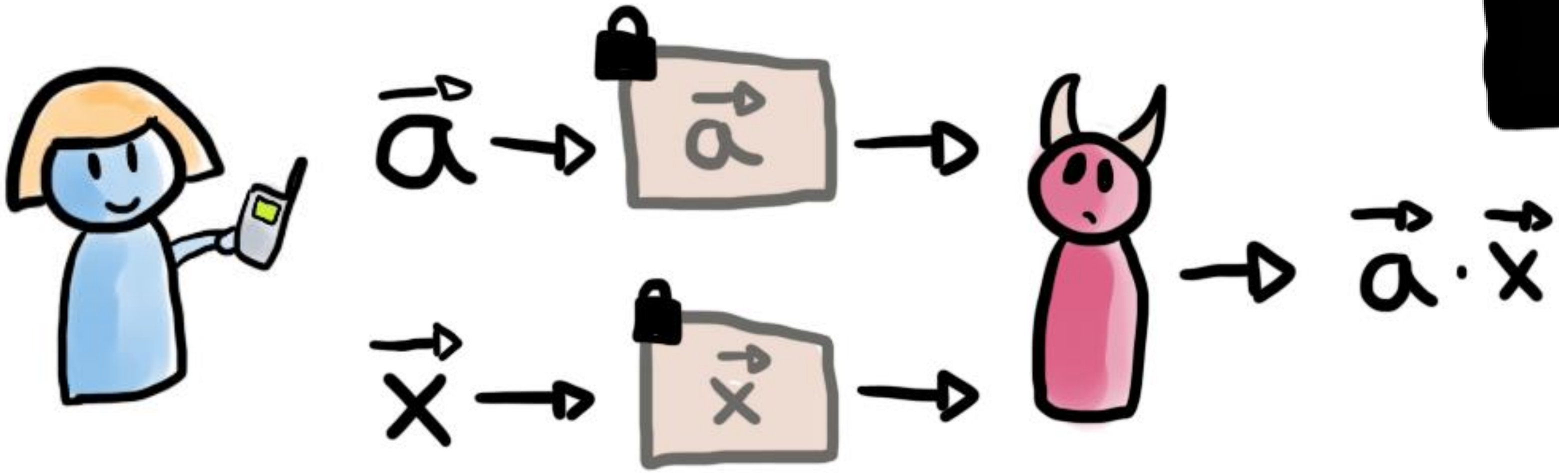


IPPE: Inner Product Predicate Encryption

[redacted]

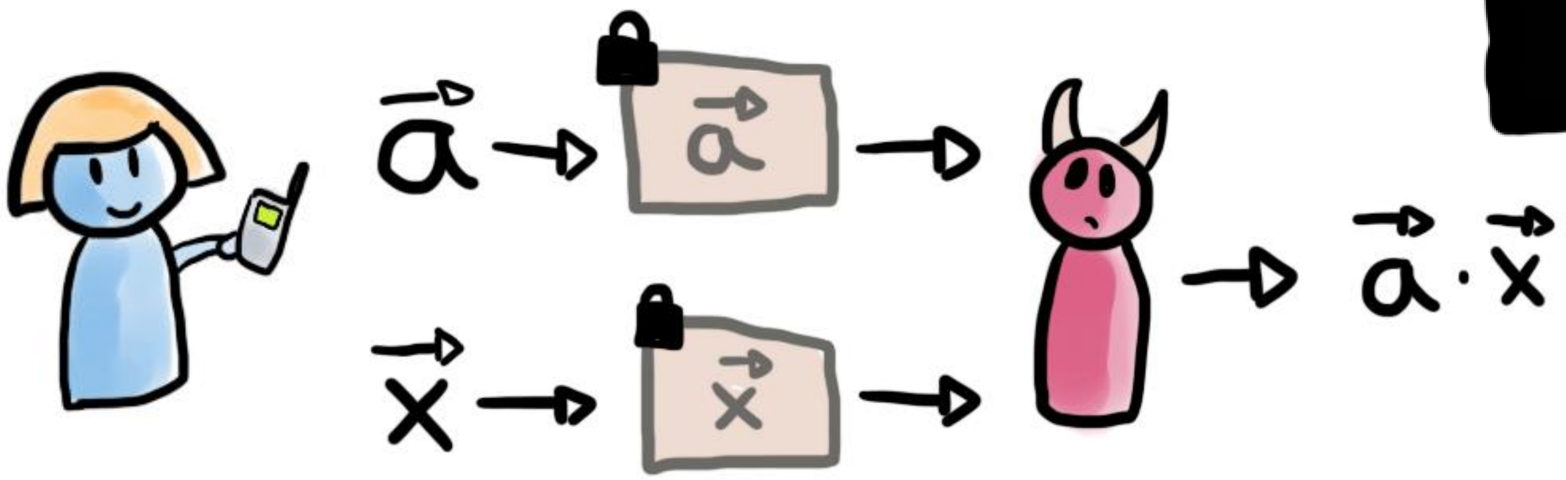
IPPE: Inner Product Predicate Encryption

[redacted]



IPPE: Inner Product Predicate Encryption

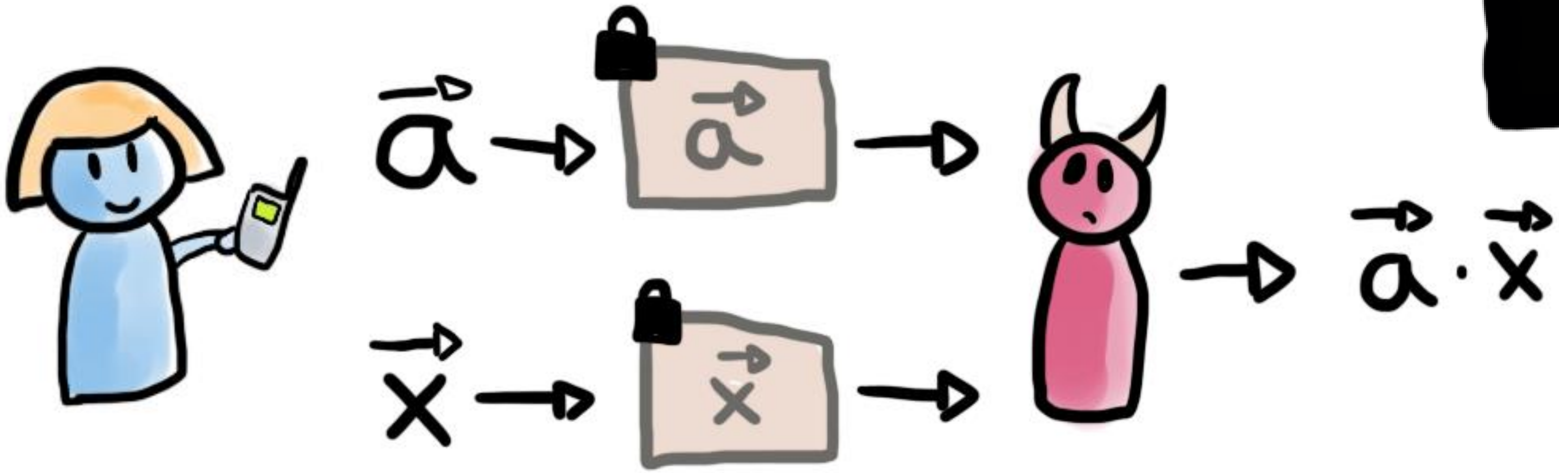
[redacted]



$$P(x) = (x - r_1)(x - r_2) \cdots (x - r_d) =$$

IPPE: Inner Product Predicate Encryption

[redacted]

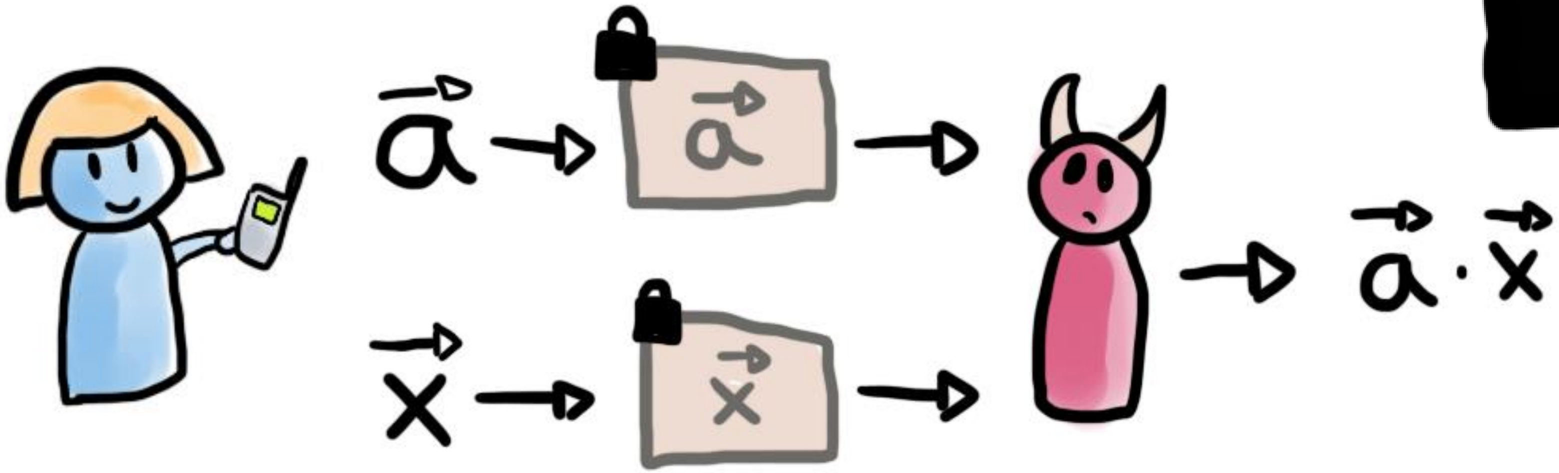


$$P(x) = (x - r_1)(x - r_2)\cdots(x - r_d) =$$

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_d \cdot x^d$$

IPPE: Inner Product Predicate Encryption

[redacted]

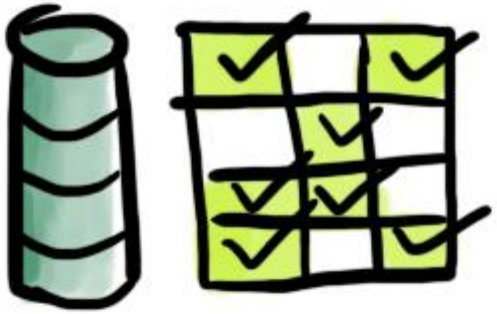


$$P(x) = (x - r_1)(x - r_2) \cdots (x - r_d) = a_0 + a_1 x + a_2 x^2 + \cdots + a_d \cdot x^d = \vec{a} \cdot \vec{x}$$

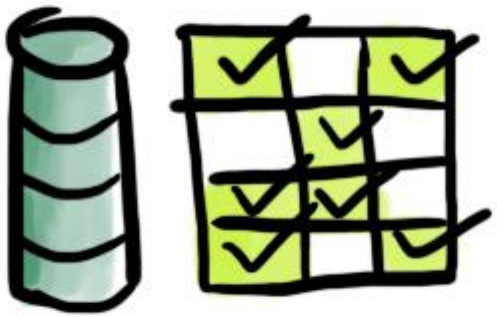
$$\vec{x} = (x^0, x^1, x^2, \ldots)$$

OSSE: Obfuscated SSE

[redacted]



OSSE: Obfuscated SSE

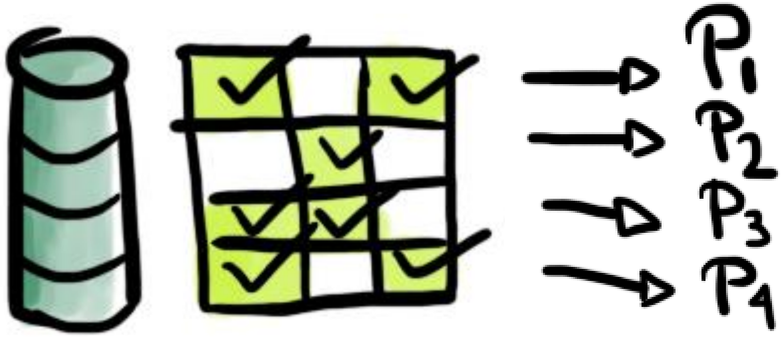


[redacted]

$h : [n] \rightarrow [|h|]$

Docs Labels

OSSE: Obfuscated SSE

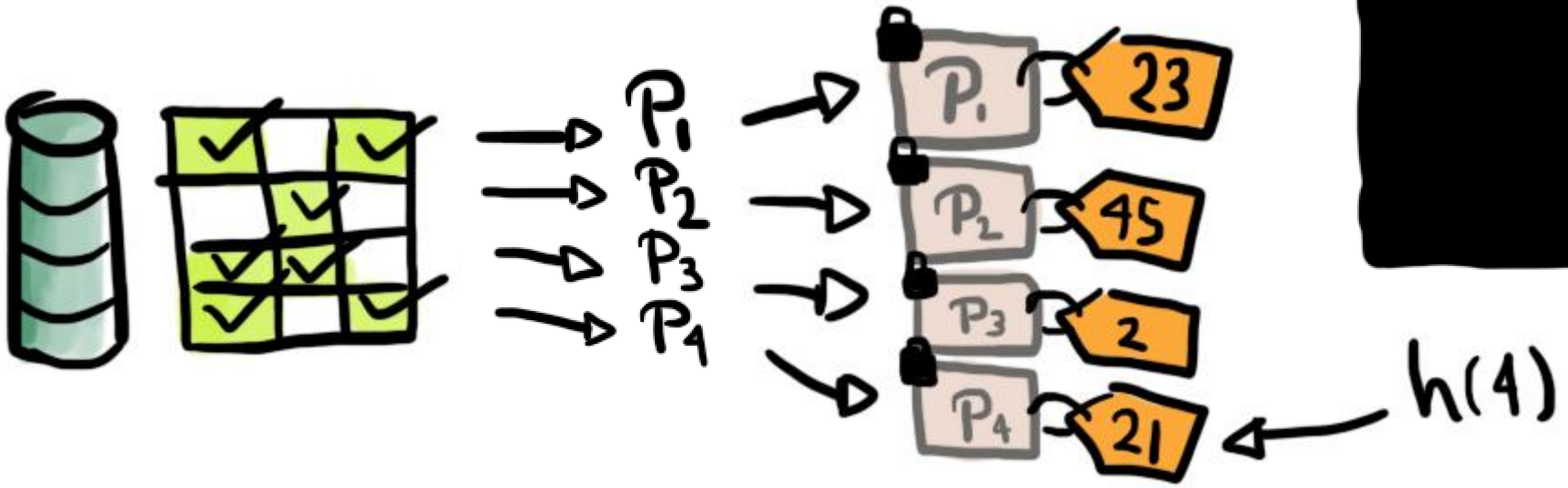


$h:[n]\to[|h|]$

Docs Labels

OSSE: Obfuscated SSE

[redacted]

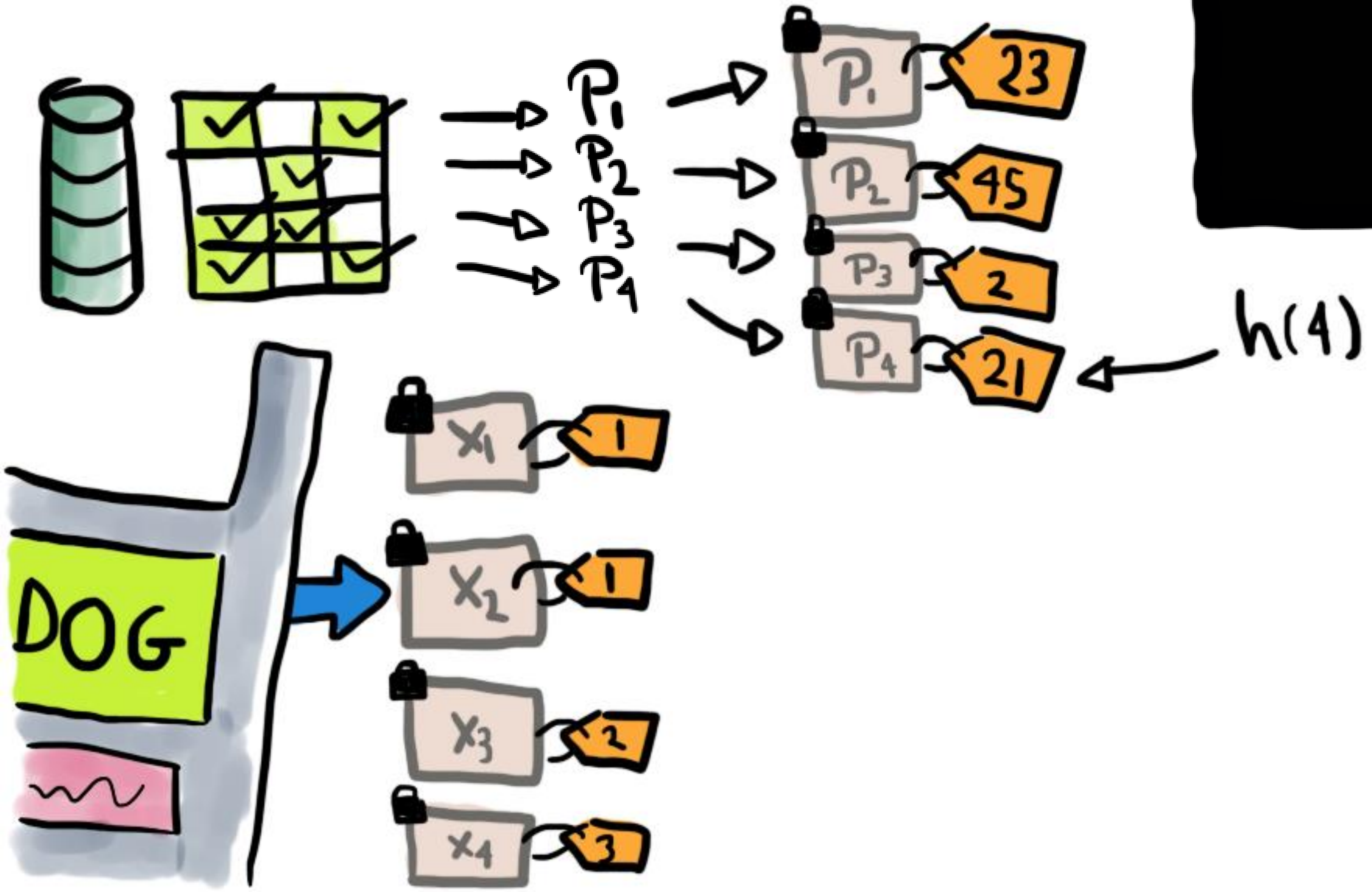


$h: [n] \to [|h|]$

Docs Labels

OSSE: Obfuscated SSE

[redacted]

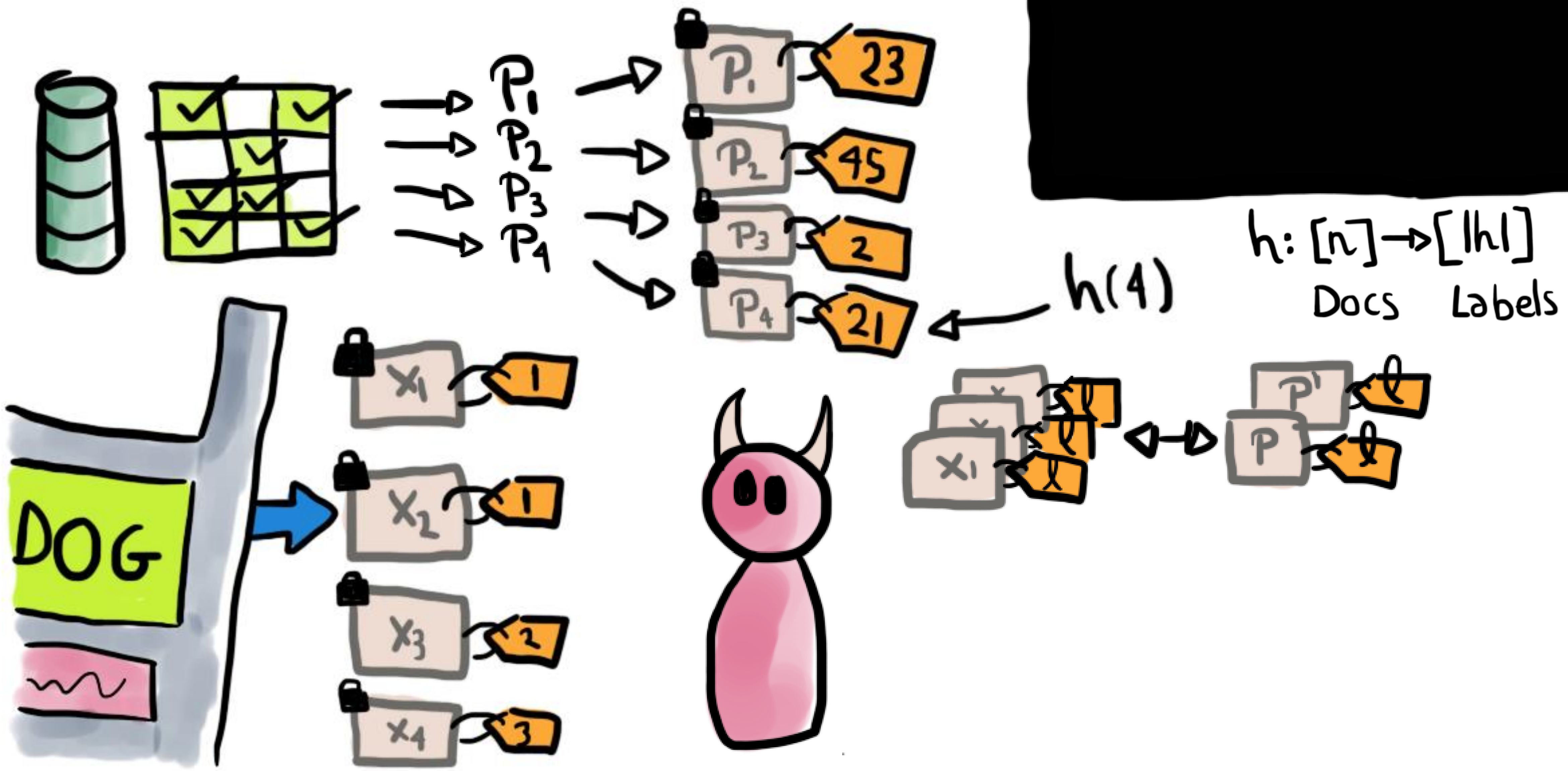


$h: [n] \to [|h|]$

Docs Labels

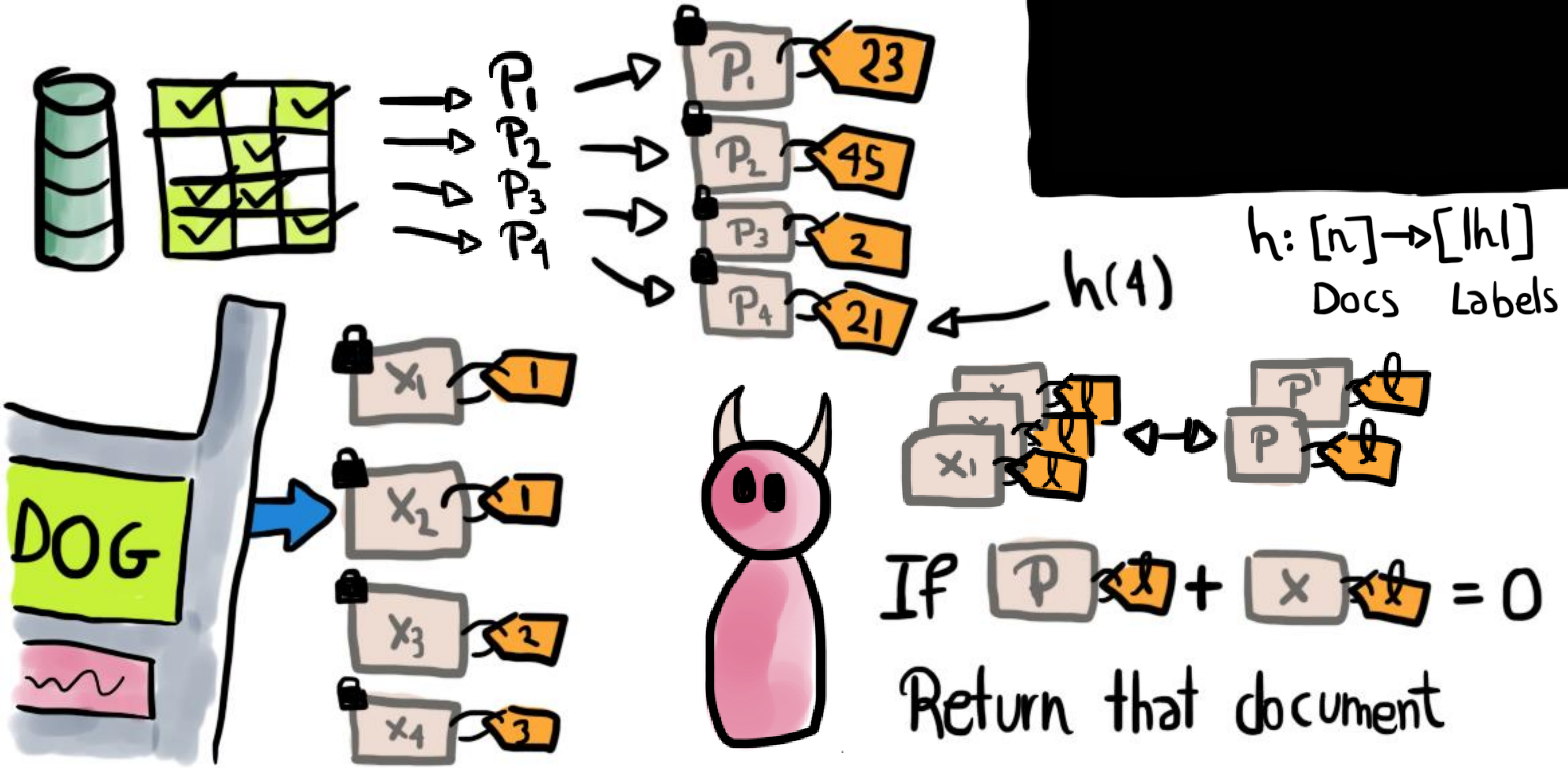
OSSE: Obfuscated SSE

[redacted]



OSSE: Obfuscated SSE

[redacted]



$h: [n] \to [|h|]$

Docs Labels

Return that document

Polynomial Generation

$$D_{30} = \{\text{DOG}, \text{COW}, \text{RAT}\}$$

$$\ell = h(30)$$

Polynomial Generation

[redacted]

$r_1 = (\text{DOG} \| \ell \| 5)$

$r_2 = (\text{COW} \| \ell \| 0)$

$r_3 = (\text{RAT} \| \ell \| 1)$

$\mathcal{D}_{30} = \{\text{DOG}, \text{COW}, \text{RAT}\}$

$\ell = h(30)$

Polynomial Generation

[redacted]

$r_1 = (\text{DOG} \| \ell \| 5)$ ← There are 5 $(\text{DOG} \| \ell \| \cdots)$ already

$r_2 = (\text{COW} \| \ell \| 0)$

$r_3 = (\text{RAT} \| \ell \| 1)$

$D_{30} = \{\text{DOG}, \text{COW}, \text{RAT}\}$

$\ell = h(30)$

Polynomial Generation

[redacted]

$r_1 = (DOG \| \ell \| 5)$ ← There are 5 $(DOG \| \ell \| \dots)$ already

$r_2 = (COW \| \ell \| 0)$

$r_3 = (RAT \| \ell \| 1)$

$D_{30} = \{DOG, COW, RAT\}$

$\ell = h(30)$

$S_{max} =$ Max keywords per document $= 5$

Polynomial Generation

[redacted]

$r_1 = (DOG \| \ell \| 5)$ ← There are 5 $(DOG\|\ell\|\cdots)$ already

$r_2 = (COW \| \ell \| 0)$

$r_3 = (RAT \| \ell \| 1)$

$r_4 = (AAA \| \ell \| 0)$

$r_5 = (AAA \| \ell \| 0)$

$$D_{30} = \{DOG, COW, RAT\}$$

$$\ell = h(30)$$

$$S_{max} = \frac{\text{Max keywords}}{\text{per document}} = 5$$

Polynomial Generation

[redacted]

$r_1 = (\text{DOG} \| \ell \| 5)$ ← There are 5 $(\text{DOG}\|\ell\|\cdots)$ already

$r_2 = (\text{COW} \| \ell \| 0)$

$r_3 = (\text{RAT} \| \ell \| 1)$

$r_4 = (\text{AAA} \| \ell \| 0)$

$r_5 = (\text{AAA} \| \ell \| 0)$

$r_6 = (30 \| 0 \| -1)$

$D_{30} = \{\text{DOG}, \text{COW}, \text{RAT}\}$

$\ell = h(30)$

$S_{max} = \dfrac{\text{Max keywords}}{\text{per document}} = 5$

Polynomial Generation

[redacted]

$r_1 = (\text{DOG} \| \ell \| 5)$ ← There are 5 $(\text{DOG}\|\ell\|\cdots)$ already

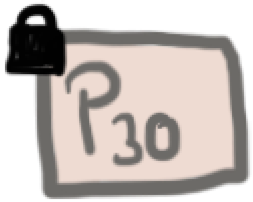
$r_2 = (\text{COW} \| \ell \| 0)$

$r_3 = (\text{RAT} \| \ell \| 1)$

$r_4 = (\text{AAA} \| \ell \| 0)$

$r_5 = (\text{AAA} \| \ell \| 0)$

$r_6 = (30 \| 0 \| -1)$



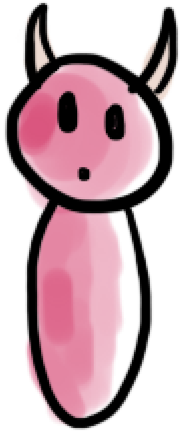
$D_{30} = \{\text{DOG}, \text{COW}, \text{RAT}\}$

$\ell = h(30)$

$S_{max} = \frac{\text{Max keywords}}{\text{per document}} = 5$

Token Generation

[redacted]



$r_1 = (DOG \| \ell \| 6)$

$r_2 = (COW \| \ell \| 0)$

$r_3 = (RAT \| \ell \| 1)$

$r_4 = (AAA \| \ell \| 0)$

$r_5 = (AAA \| \ell \| 0)$

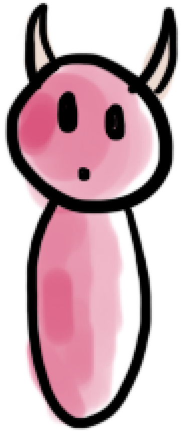
$r_6 = (30 \| 0 \| -1)$

Token Generation

[redacted]

▶ Find 🛢 with "DOG":

```
For $\ell = 1 \to |h|$:
    For $c = 0 \to c_{max}$:
        $x = (DOG \| \ell \| c) \to$
```



$r_1 = (DOG \| \ell \| 6)$

$r_2 = (COW \| \ell \| 0)$

$r_3 = (RAT \| \ell \| 1)$

$r_4 = (AAA \| \ell \| 0)$

$r_5 = (AAA \| \ell \| 0)$

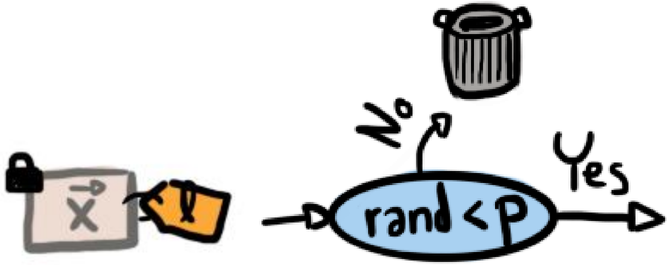
$r_6 = (30 \| 0 \| -1)$

Token Generation

[redacted]

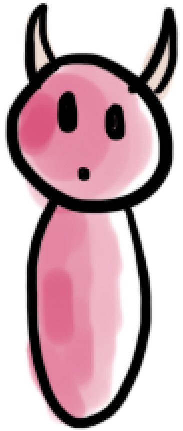
▶ Find 🛢 with "DOG":

```
For $\ell=1 \rightarrow |h|$:
    For $c=0 \rightarrow c_{max}$:
        $x=(DOG||\ell||c) \rightarrow$
```



$$r_1 = (DOG || \ell || 6)$$
$$r_2 = (COW || \ell || 0)$$
$$r_3 = (RAT || \ell || 1)$$
$$r_4 = (AAA || \ell || 0)$$
$$r_5 = (AAA || \ell || 0)$$

$$r_6 = (30 || 0 || -1)$$

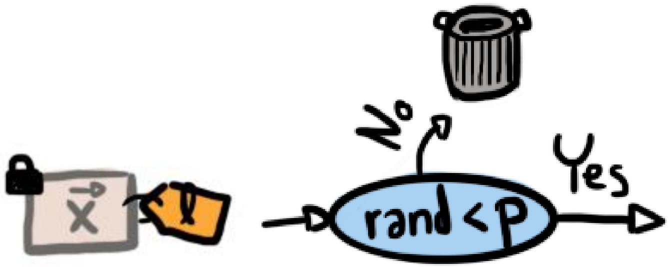


Token Generation

[redacted]

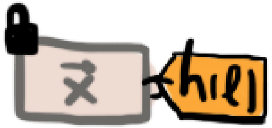
- Find ⛁ with "DOG":

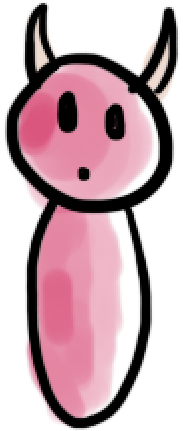
```
For $\ell = 1 \to |h|$:
  For $c = 0 \to c_{max}$:
    $x = (DOG \| \ell \| c) \to$
```



- False positives:

```
For $id = 1 \to n$:
  $x = (id \| 0 \| -1) \to$
```





$$r_1 = (DOG \| \ell \| 6)$$
$$r_2 = (COW \| \ell \| 0)$$
$$r_3 = (RAT \| \ell \| 1)$$
$$r_4 = (AAA \| \ell \| 0)$$
$$r_5 = (AAA \| \ell \| 0)$$

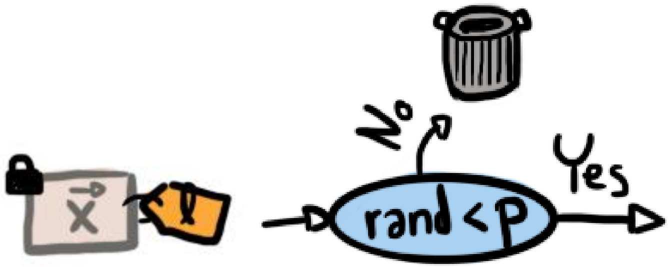
$$r_6 = (30 \| 0 \| -1)$$

Token Generation

[redacted]

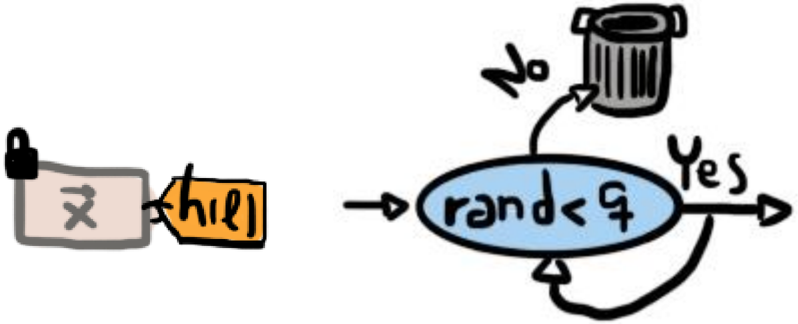
- ▶ Find ⛁ with "DOG":

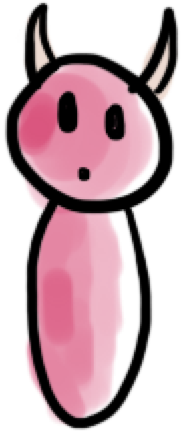
```
For $\ell = 1 \to |h|$:
    For $c = 0 \to c_{max}$:
        $x = (DOG \| \ell \| c) \to$
```



- ▶ False positives:

```
For $id = 1 \to n$:
    $x = (id \| 0 \| -1) \to$
```





$r_1 = (DOG \| \ell \| 6)$

$r_2 = (COW \| \ell \| 0)$

$r_3 = (RAT \| \ell \| 1)$

$r_4 = (AAA \| \ell \| 0)$

$r_5 = (AAA \| \ell \| 0)$

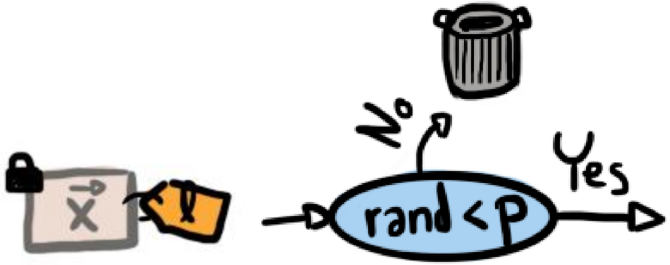
$r_6 = (30 \| 0 \| -1)$

Token Generation

[redacted]

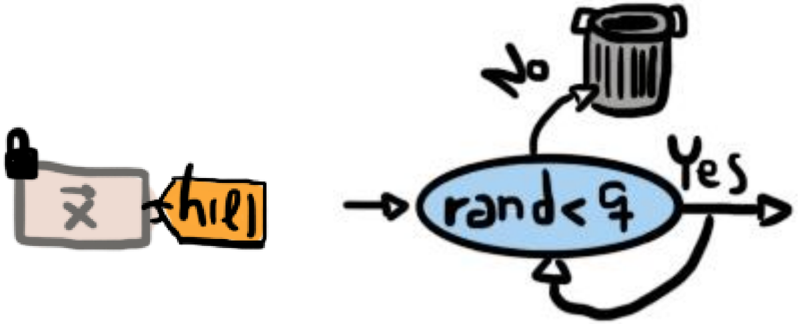
▶ Find 🛢 with "DOG":

```
For $\ell = 1 \to |h|$:
    For $c = 0 \to c_{max}$:
        $x = (DOG \| \ell \| c) \to$
```



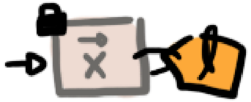
▶ False positives:

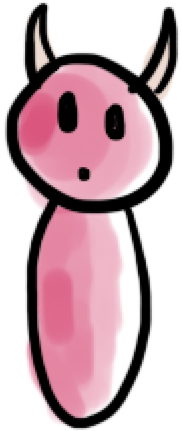
```
For $id = 1 \to n$:
    $x = (id \| 0 \| -1) \to$
```



▶ Non-matches:

```
For $\ell = 1 \to |h|$:
    $x = (AAA \| -1 \| 0) \to$
```





$$r_1 = (DOG \| \ell \| 6)$$

$$r_2 = (COW \| \ell \| 0)$$

$$r_3 = (RAT \| \ell \| 1)$$

$$r_4 = (AAA \| \ell \| 0)$$

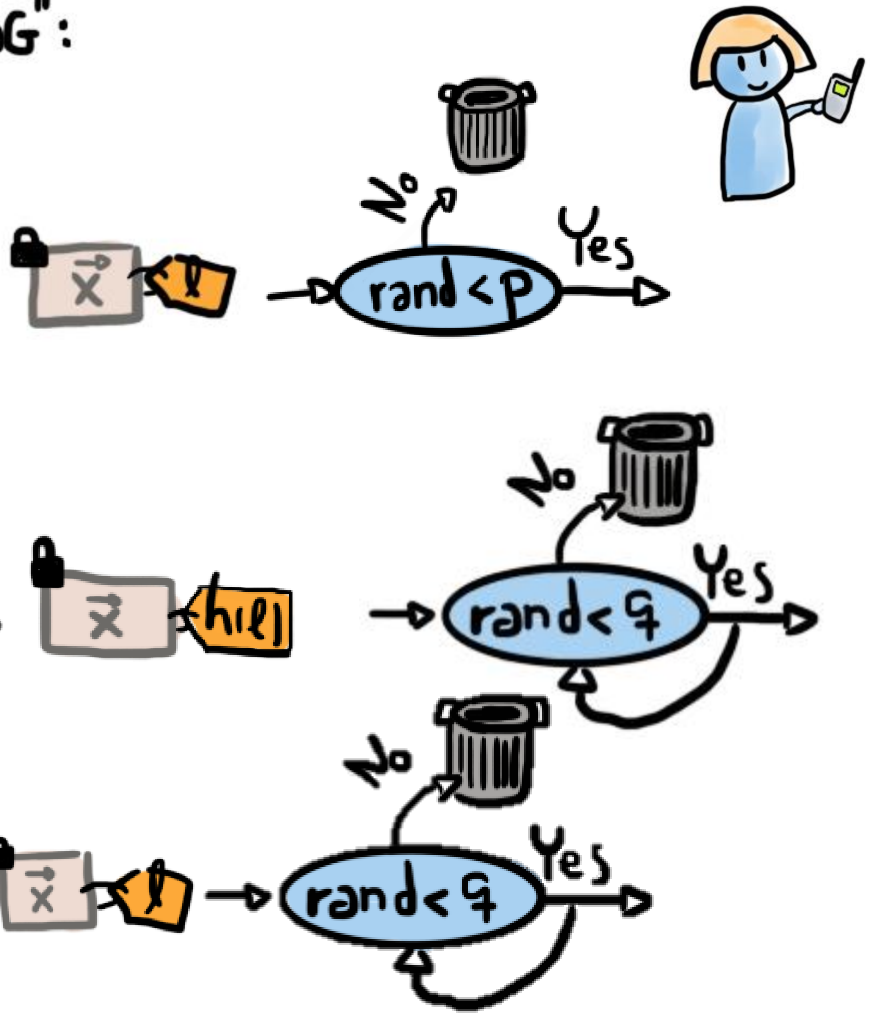
$$r_5 = (AAA \| \ell \| 0)$$

$$r_6 = (30 \| 0 \| -1)$$

Token Generation

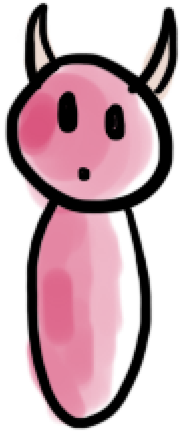
[redacted]

```
▶ Find ⛁ with "DOG":
  For $\ell = 1 \to |h|$:
    For $c = 0 \to c_{max}$:
      $x = (DOG || \ell || c) \to$
```



```
▶ False positives:
  For $id = 1 \to n$:
    $x = (id || 0 || -1) \to$

▶ Non-matches:
  For $\ell = 1 \to |h|$:
    $x = (AAA || -1 || 0) \to$
```



$r_1 = (DOG || \ell || 6)$

$r_2 = (COW || \ell || 0)$

$r_3 = (RAT || \ell || 1)$

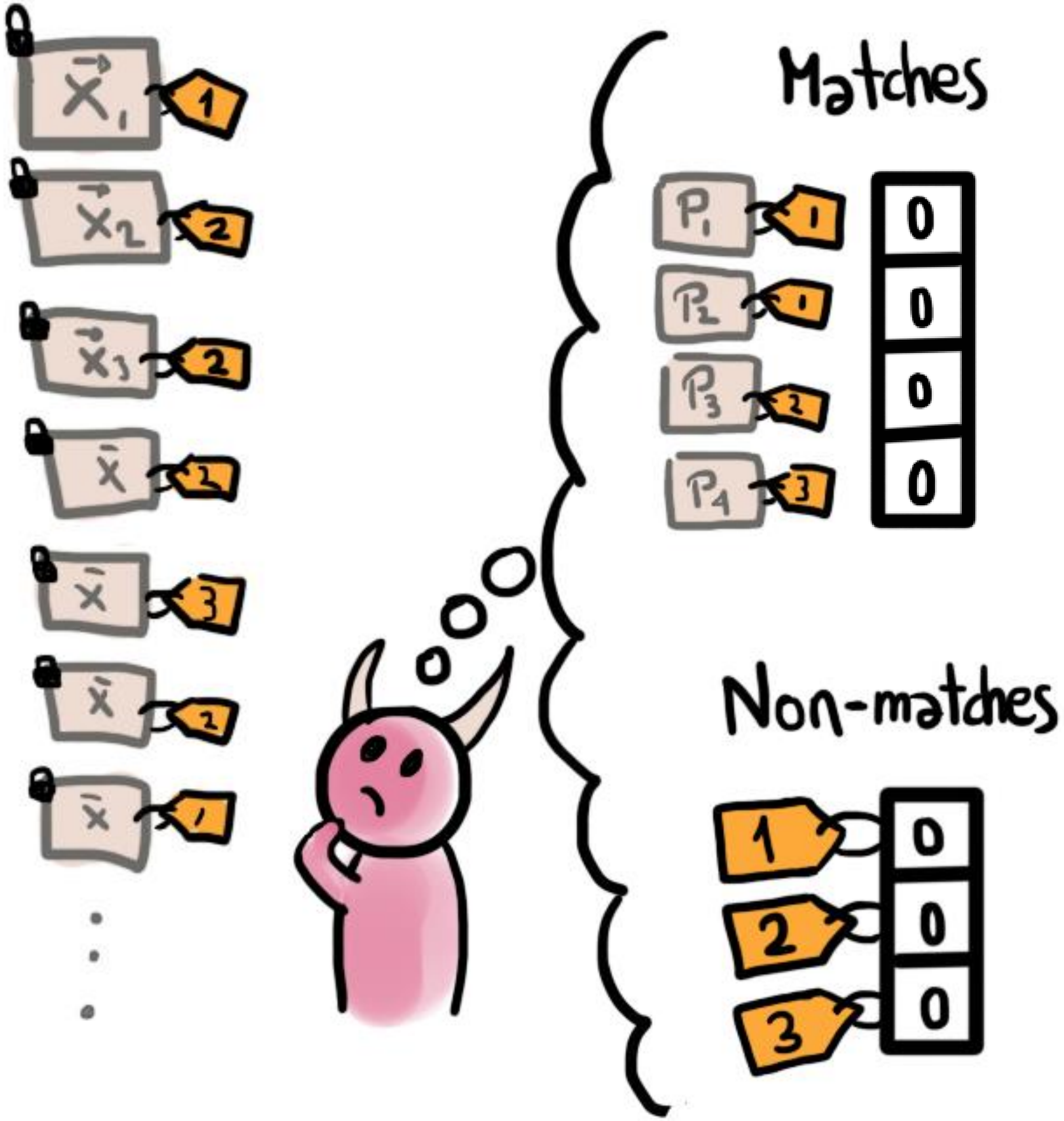
$r_4 = (AAA || \ell || 0)$

$r_5 = (AAA || \ell || 0)$

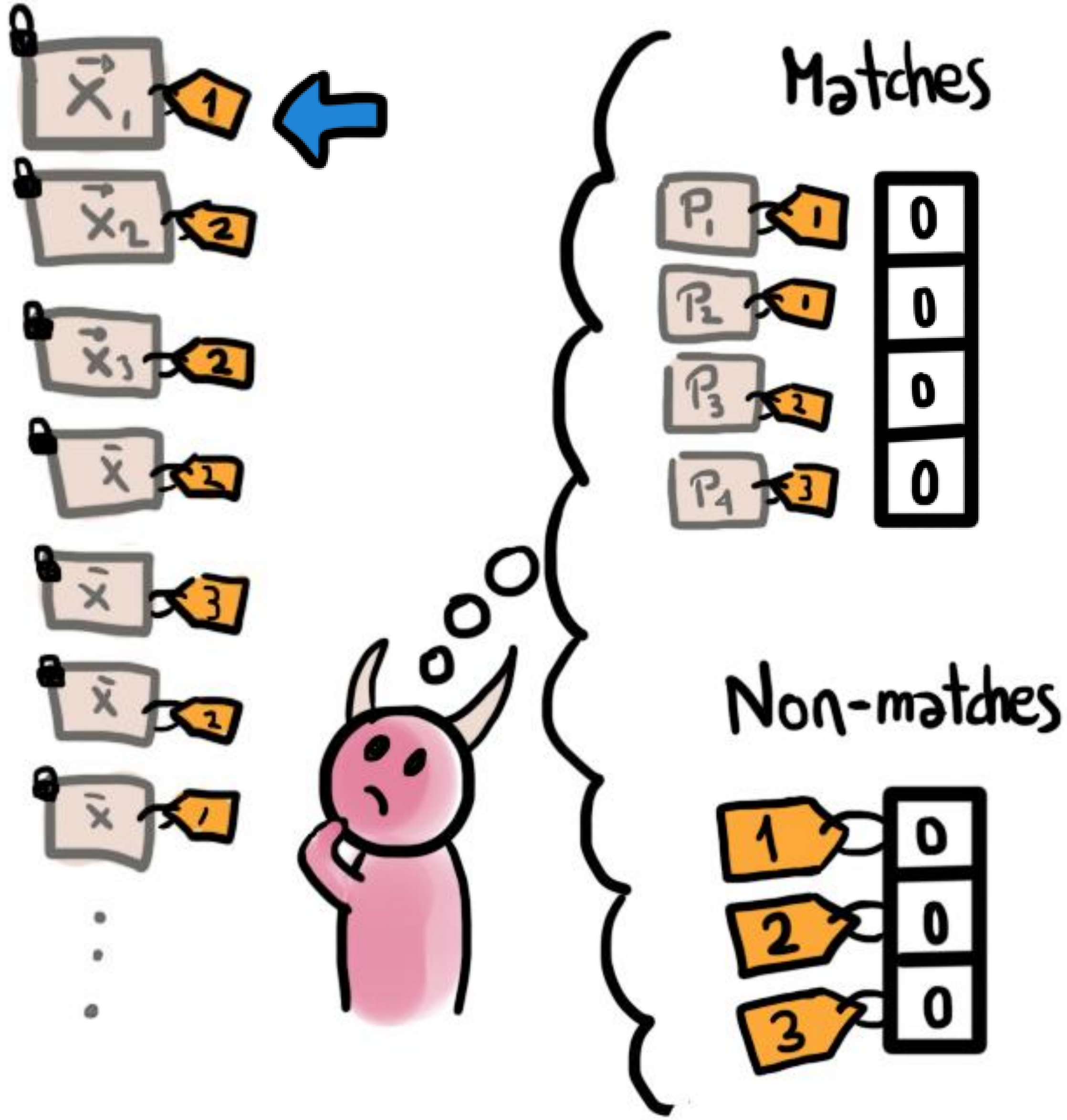
$r_6 = (30 || 0 || -1)$

Adversary's View

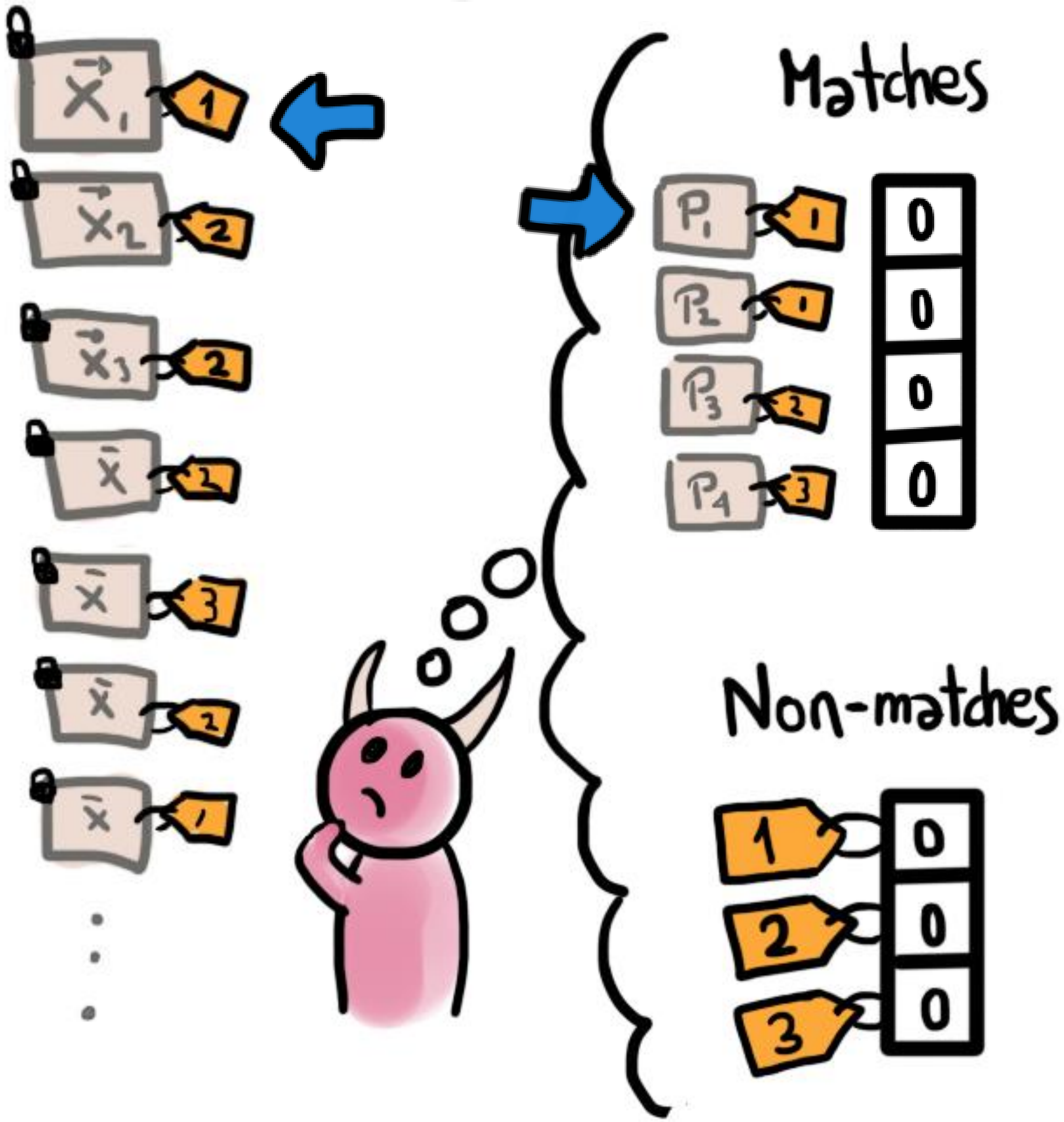
[redacted]



Adversary's View



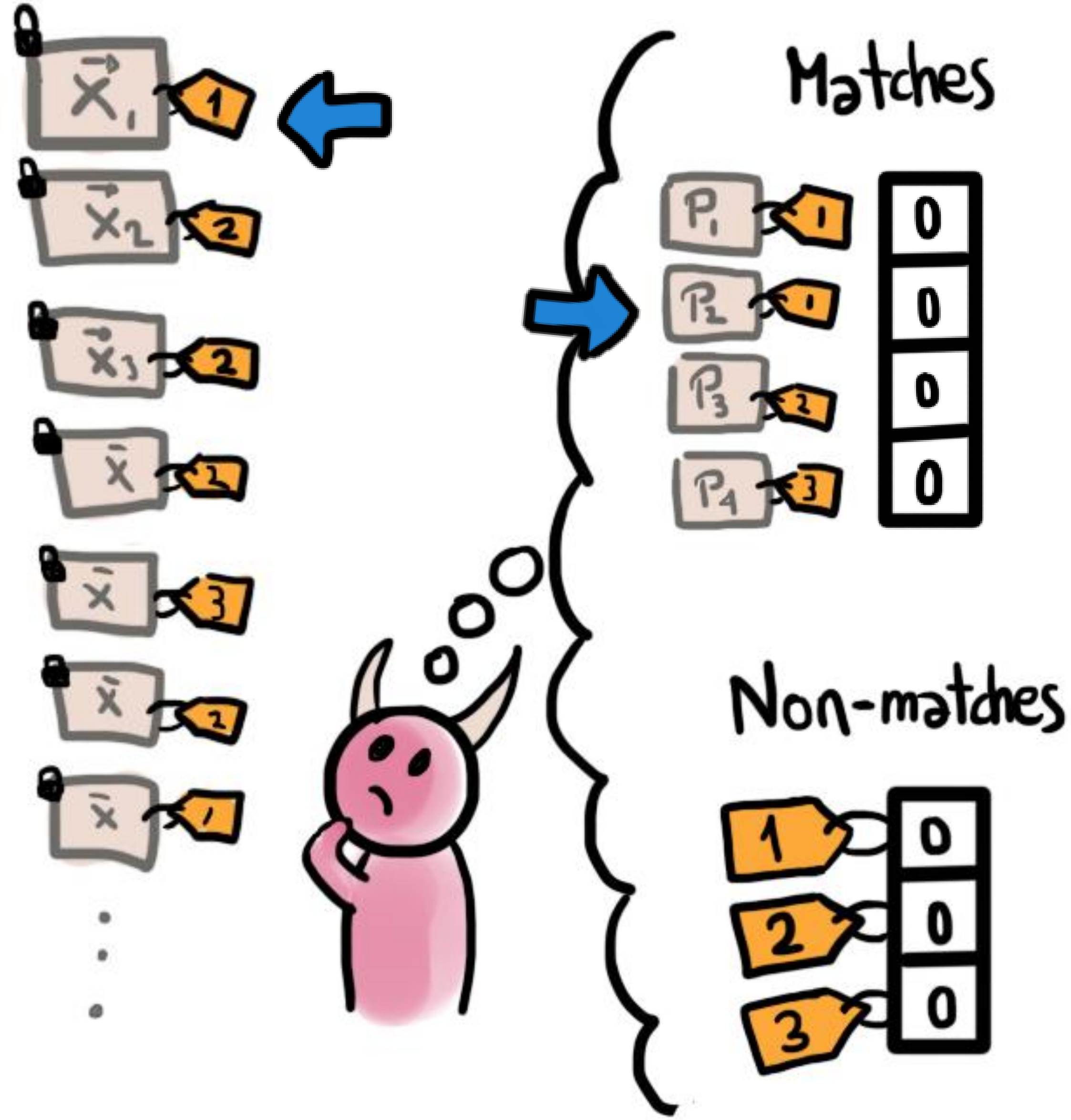
Adversary's View DOG



[redacted]

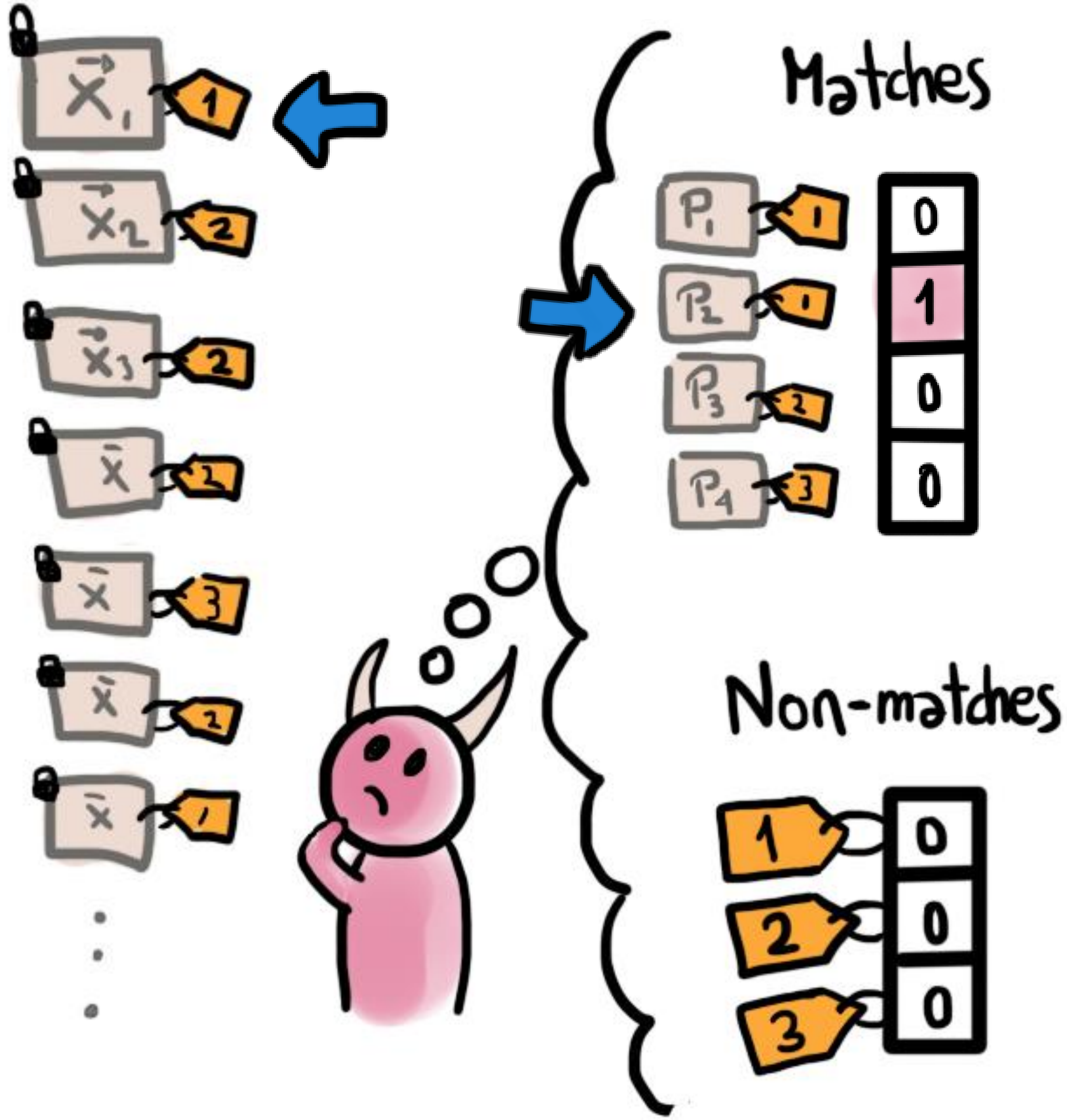
Adversary's View DOG

[redacted]

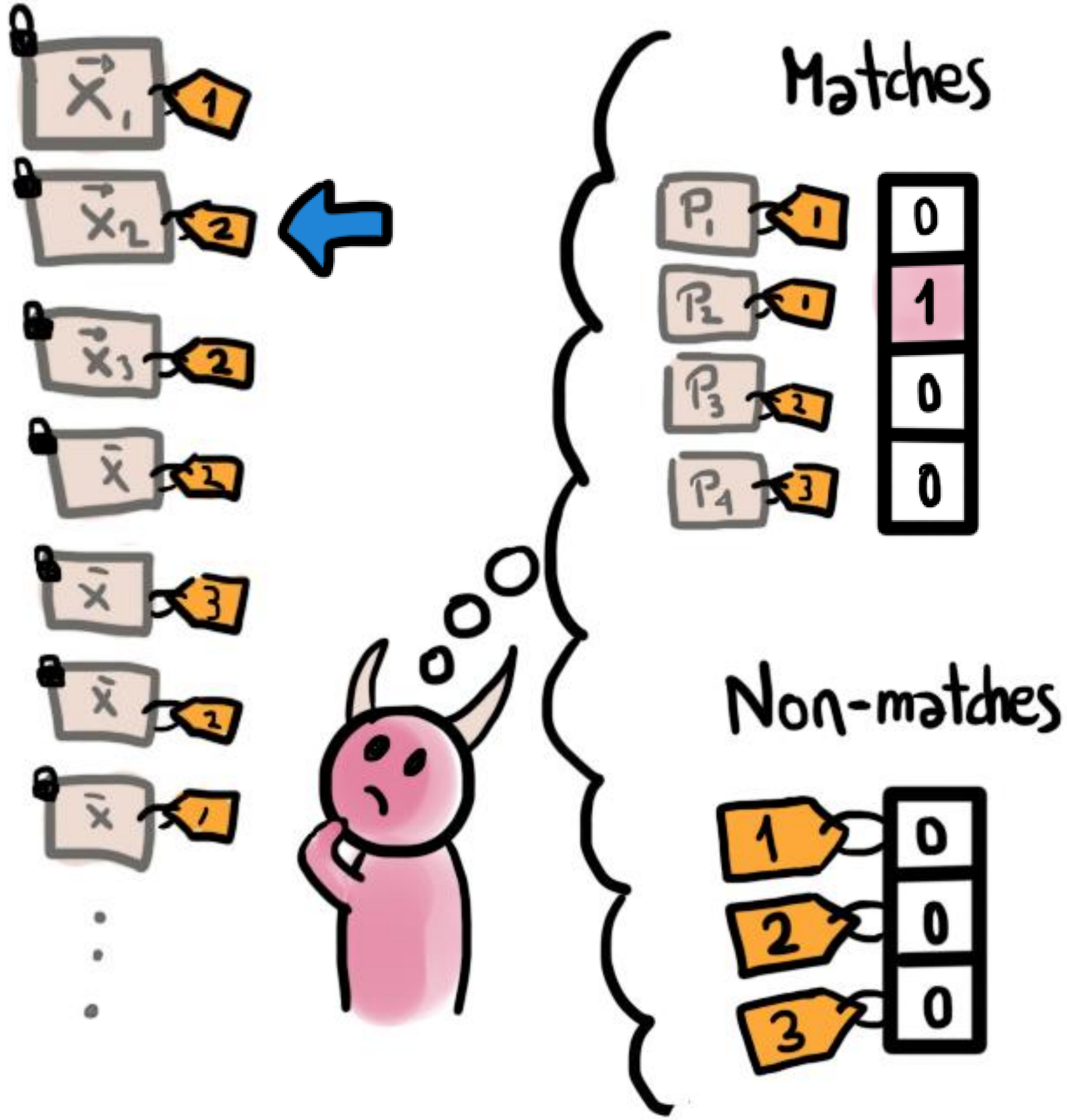


Adversary's View

[redacted]

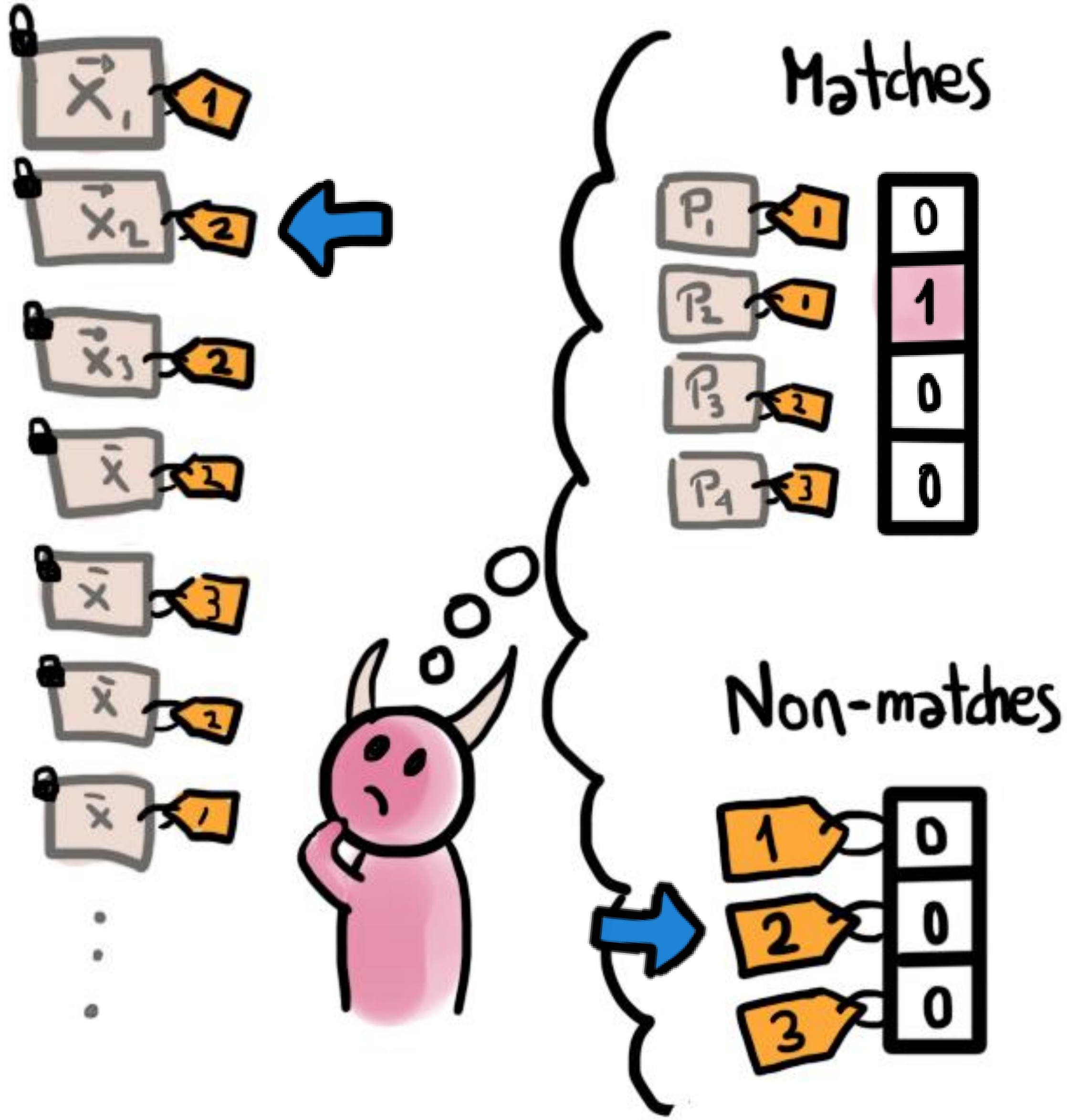


Adversary's View DOG

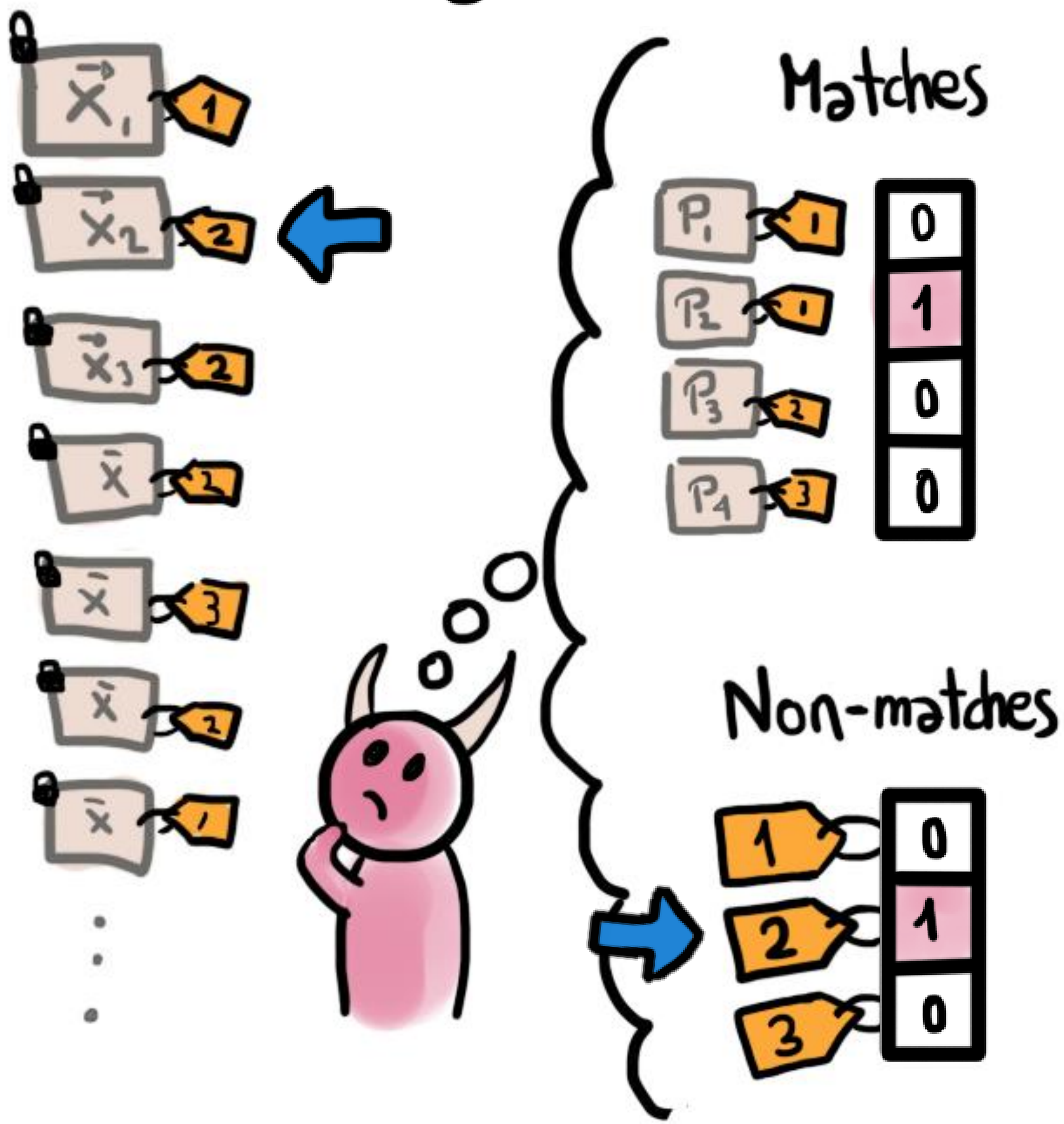


[redacted]

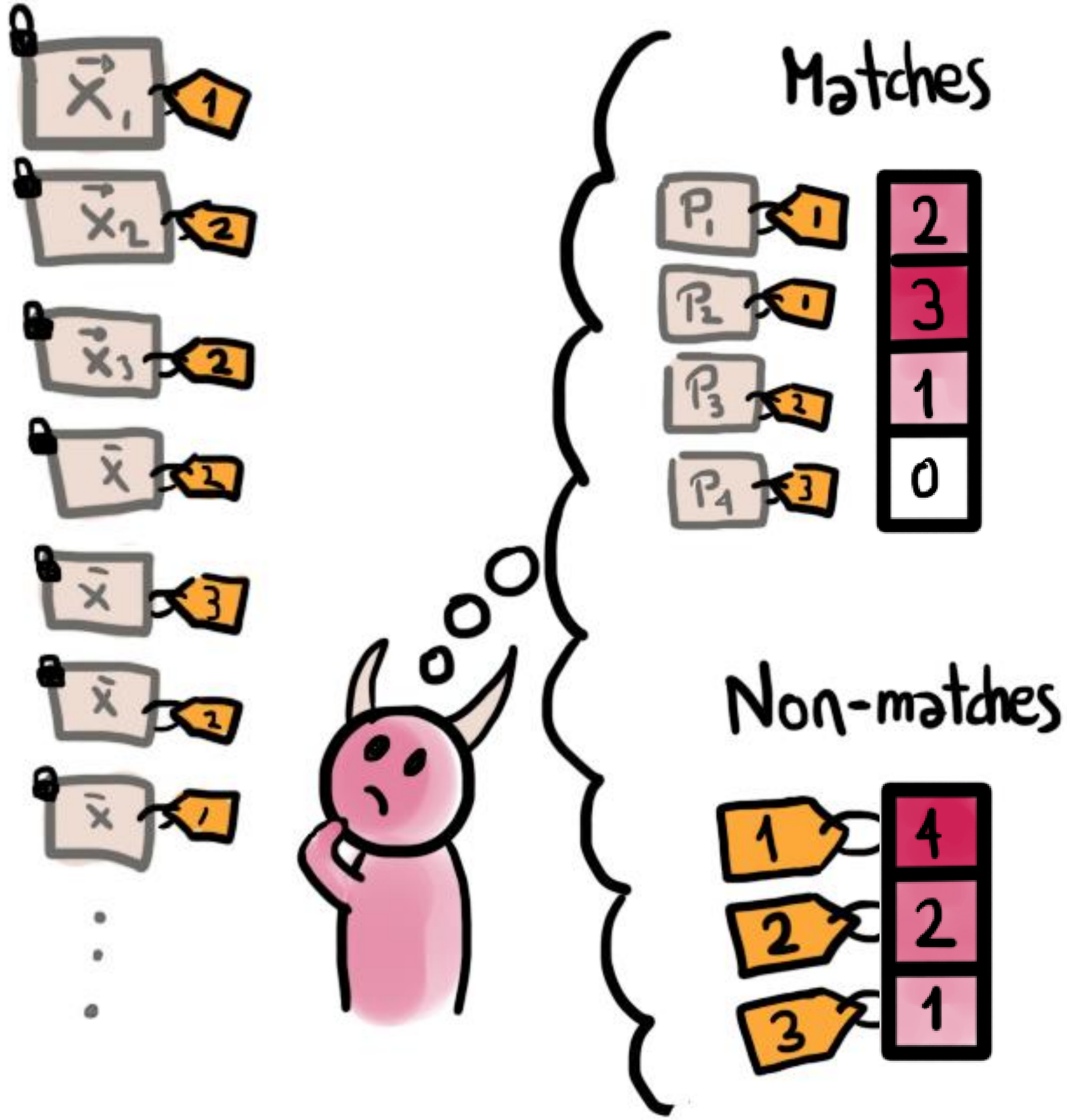
Adversary's View DOG



Adversary's View



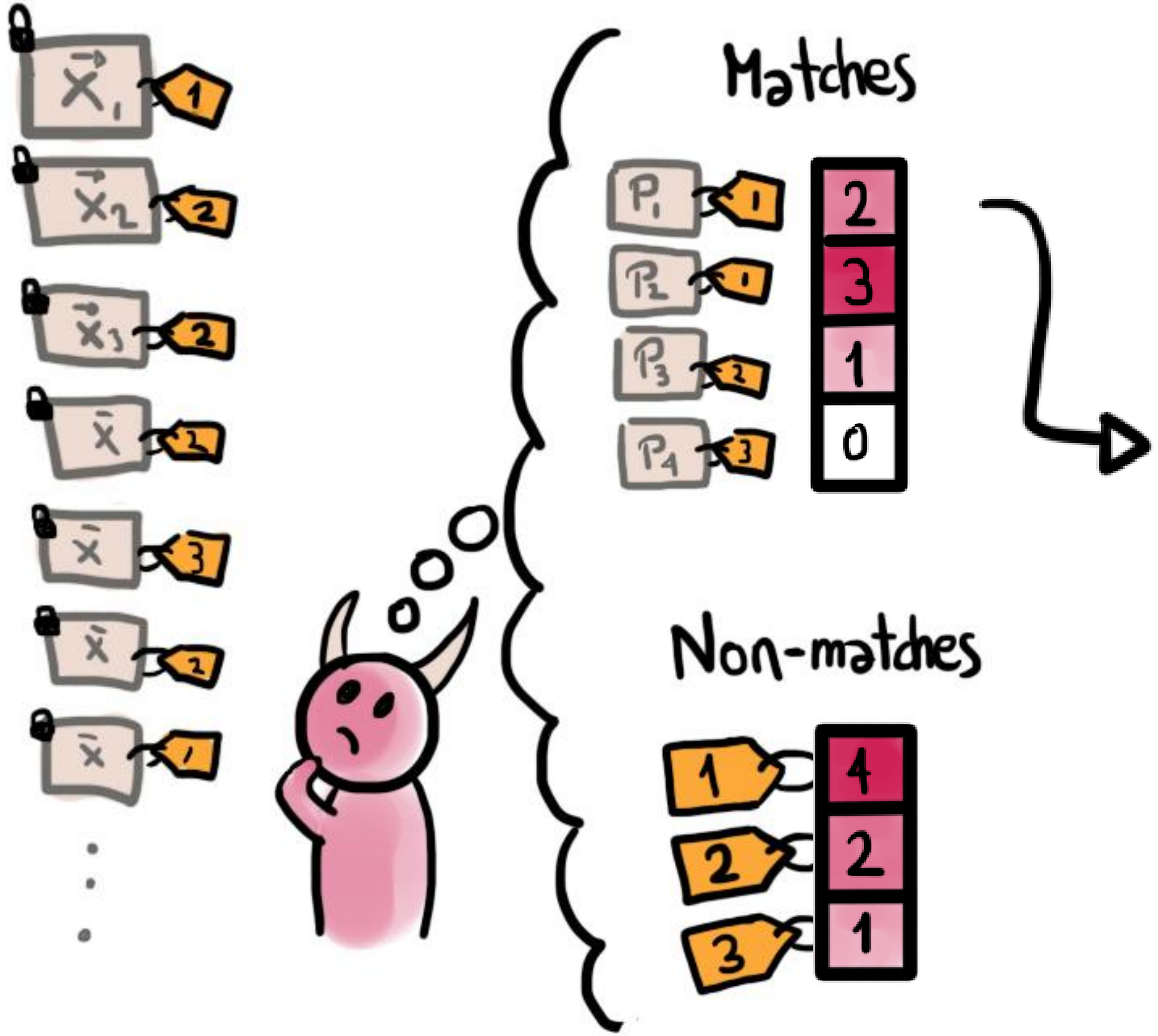
Adversary's View DOG



[redacted]

Adversary's View

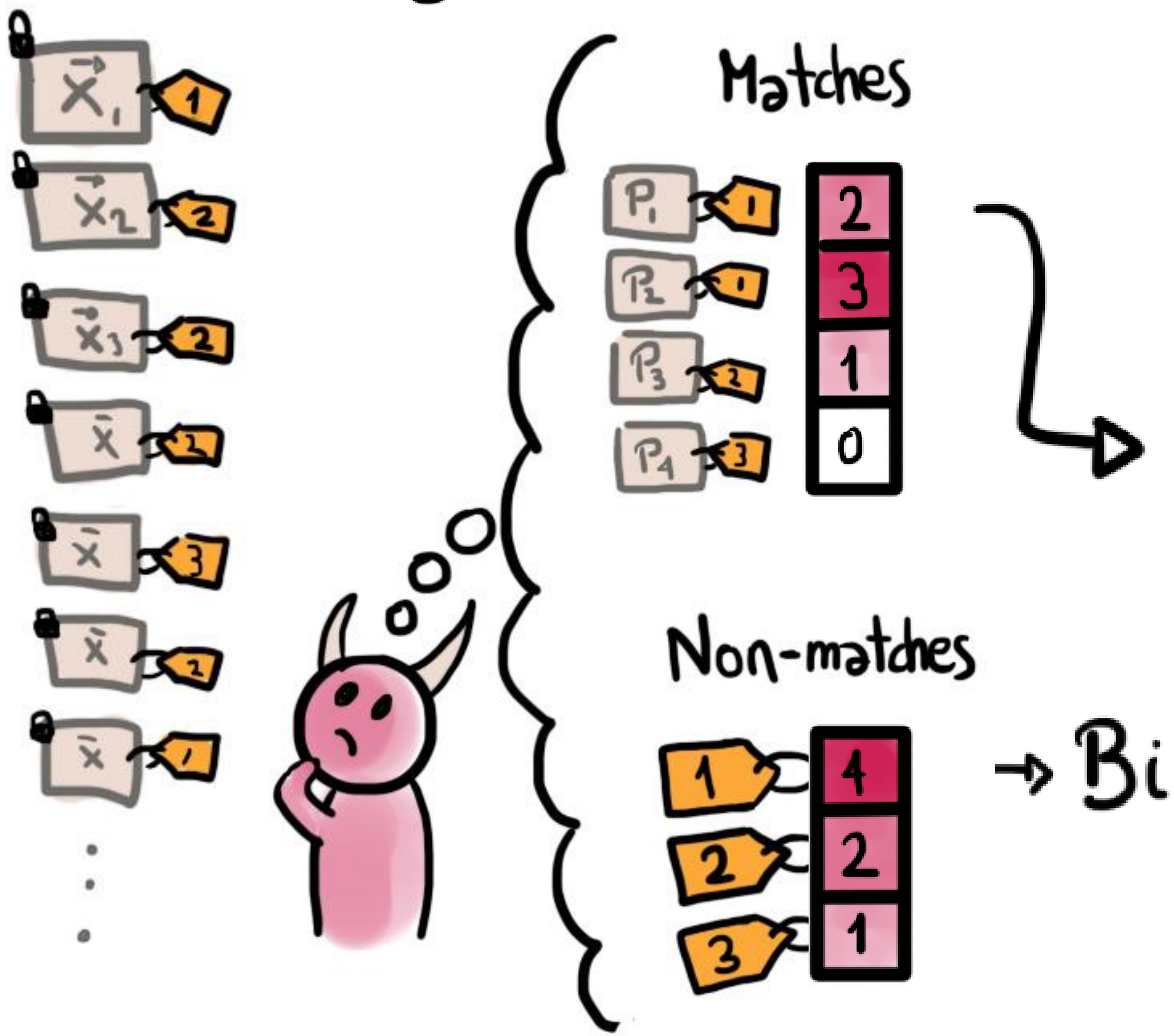
[redacted]



$$\begin{cases} \mathrm{Ber}(p) + \mathrm{Geo}(1-q) & \mathrm{DOG} \in D_1 \\ \mathrm{Geo}(1-q) & \mathrm{DOG} \notin D \end{cases}$$

Adversary's View

[redacted]

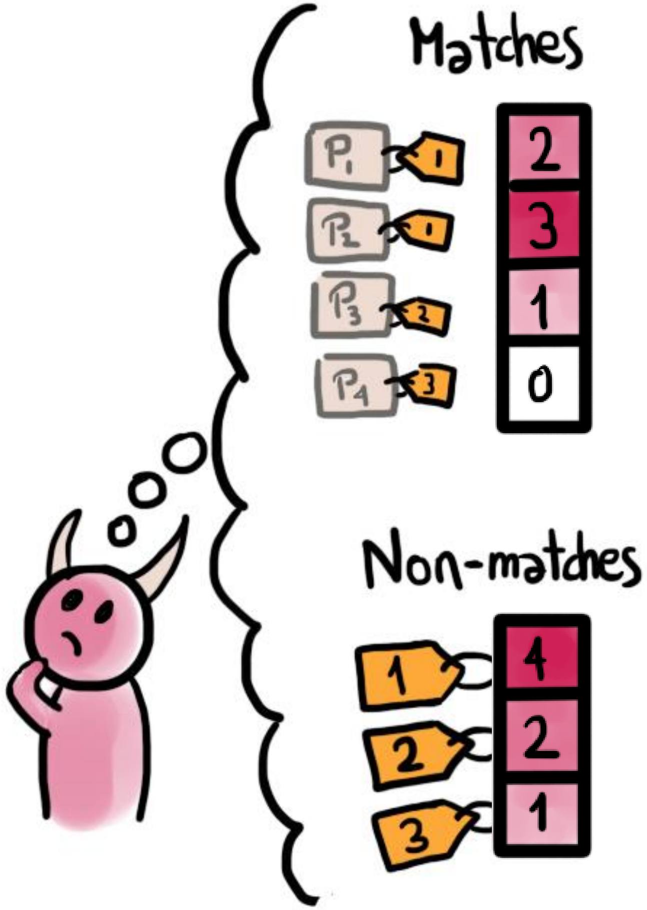


$$\begin{cases} \mathrm{Ber}(p) + \mathrm{Geo}(1-q) & \mathrm{DOG} \in D_1 \\ \mathrm{Geo}(1-q) & \mathrm{DOG} \notin D \end{cases}$$

$\rightarrow \mathrm{Bi}(g_1, p) + \mathrm{Geo}(1-q)$

Security

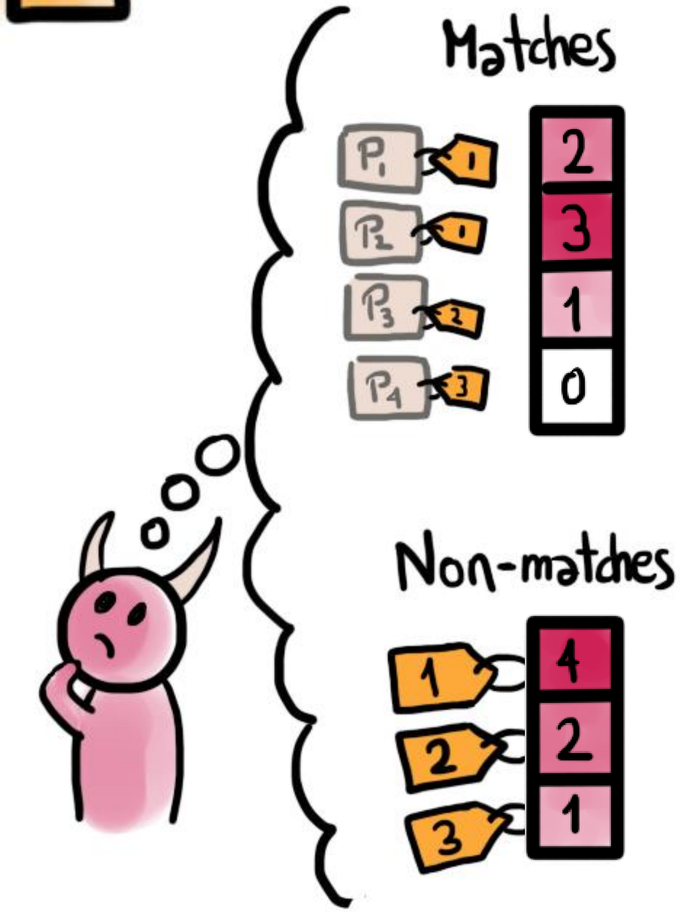
[redacted]



Security

[redacted]

We prove it holds by IPPE security



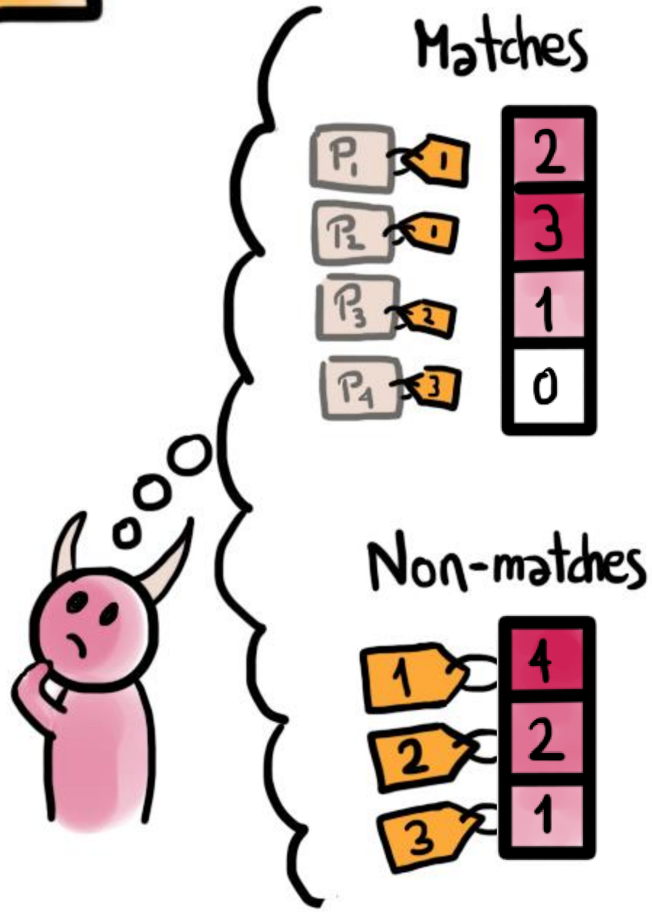
[redacted]

Security

We prove it holds by IPPE security

Differential Privacy

$$\varepsilon = \ln\left(\frac{TPR}{FPR} \cdot \frac{1-FPR}{1-TPR}\right)$$



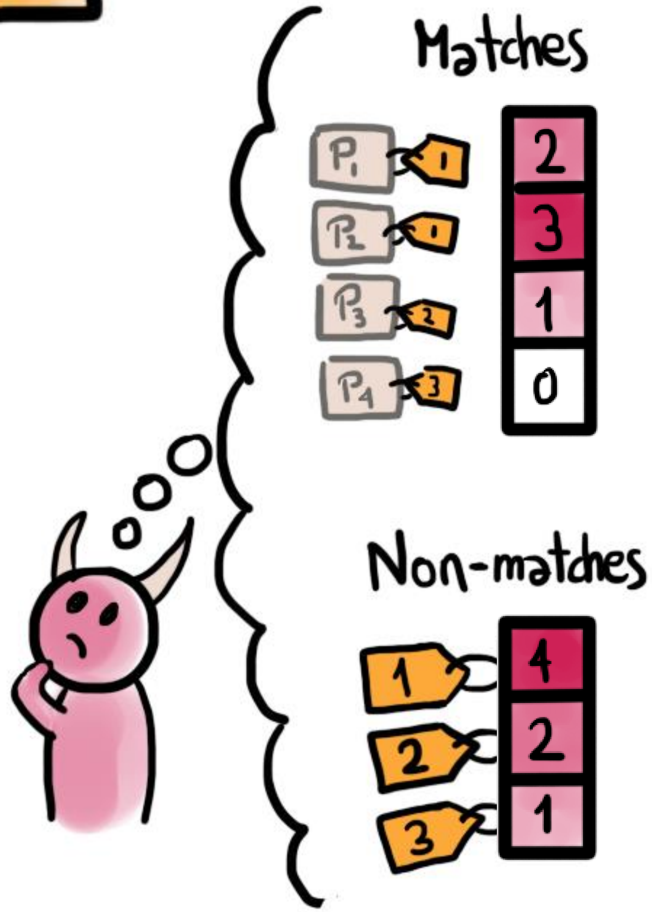
Security

We prove it holds by IPPE security

[redacted]

Differential Privacy

$$\varepsilon = \ln\left(\frac{TPR}{FPR} \cdot \frac{1-FPR}{1-TPR}\right)$$



$$\left.\begin{array}{l} TPR = 0.9999 \\ FPR = 0.025 \end{array}\right\} \varepsilon = 13$$

Complexity Analysis

Complexity Analysis

- Communication overhead (Zipf)

$$\text{COMM} = O(\log n_{\text{keywords}})$$

1 round

[redacted]

Complexity Analysis

[redacted]

- Communication overhead (Zipf)

  $COMM = O(\log n_{keywords})$

  1 round

- Computational complexity

  $COMP < n \cdot (C_{max} + 1)$

Complexity Analysis

[redacted]

- Communication overhead (Zipf)

$COMM = O(\log n_{keywords})$

1 round

- Computational complexity

$COMP < n \cdot (c_{max} + 1)$

- Client Storage:

Complexity Analysis

- Communication overhead (Zipf)

  $COMM = O(\log n_{keywords})$

  1 round

- Computational complexity

  $COMP < n \cdot (c_{max} + 1)$

- Client Storage:

TWORAM (ORAM)

$O(\log n \cdot \log\log n)$

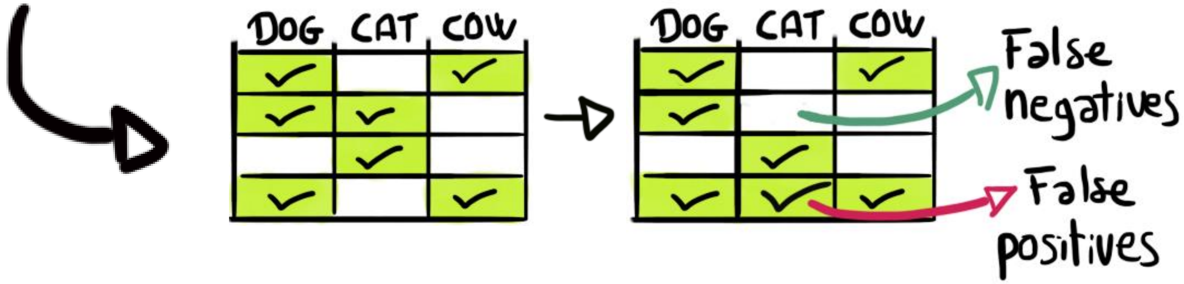
4 rounds at least

$O(\log^2 n)$ storage

Evaluation:

→ CLRZ vs. OSSE

[redacted]



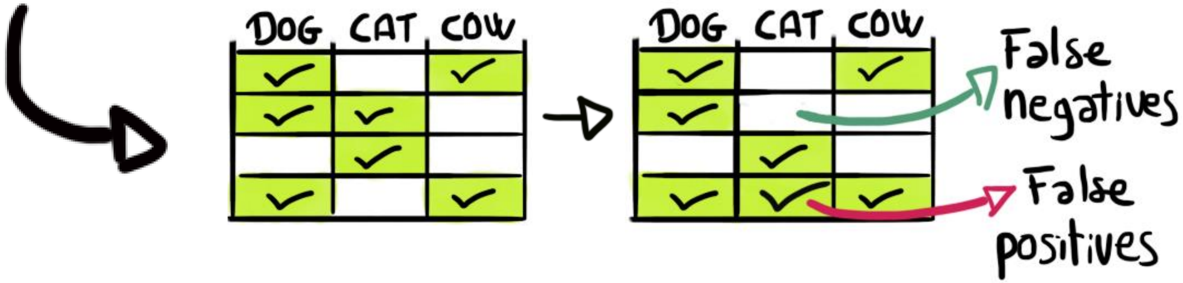
DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

Evaluation:

[redacted]

- → CLRZ vs. OSSE



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

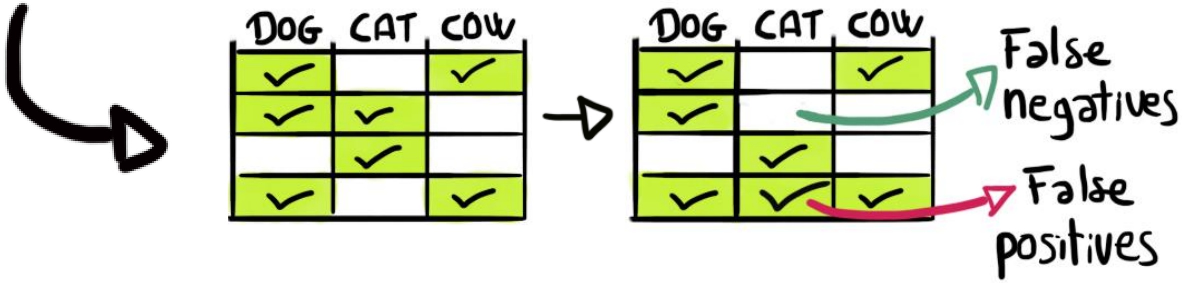
DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

- → Four different query recovery attacks

Evaluation:

[redacted]

- CLRZ vs. OSSE



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

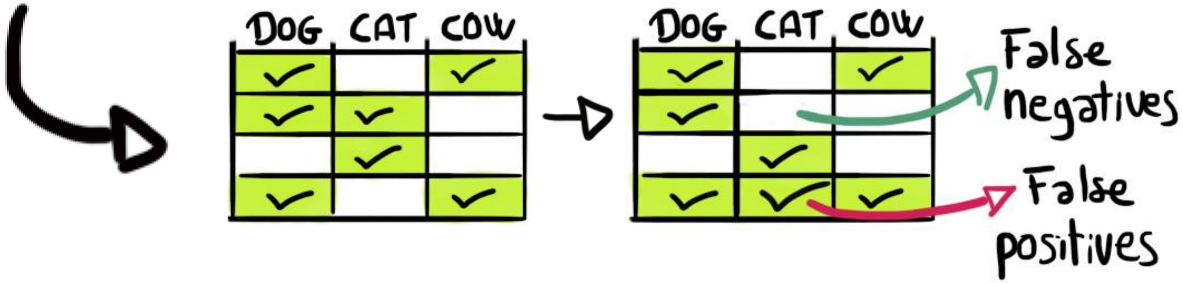
DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

- Four different query recovery attacks
- Enron dataset

Evaluation:

[redacted]

- CLRZ vs. OSSE



DOG	CAT	COW
✓		✓
✓	✓	
	✓	
✓		✓

DOG	CAT	COW
✓		✓
✓		
	✓	
✓	✓	✓

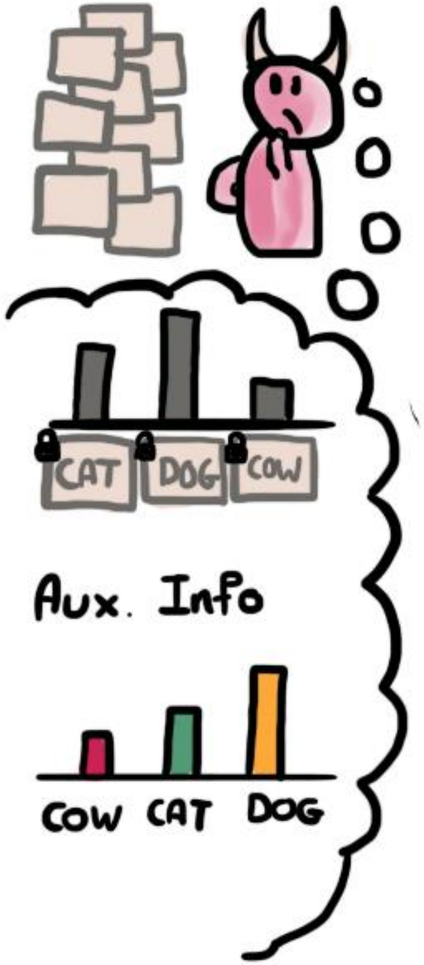
- Four different query recovery attacks
- Enron dataset
- We adapt the attacks against the defenses

Evaluation: Frequency Attack

[redacted]

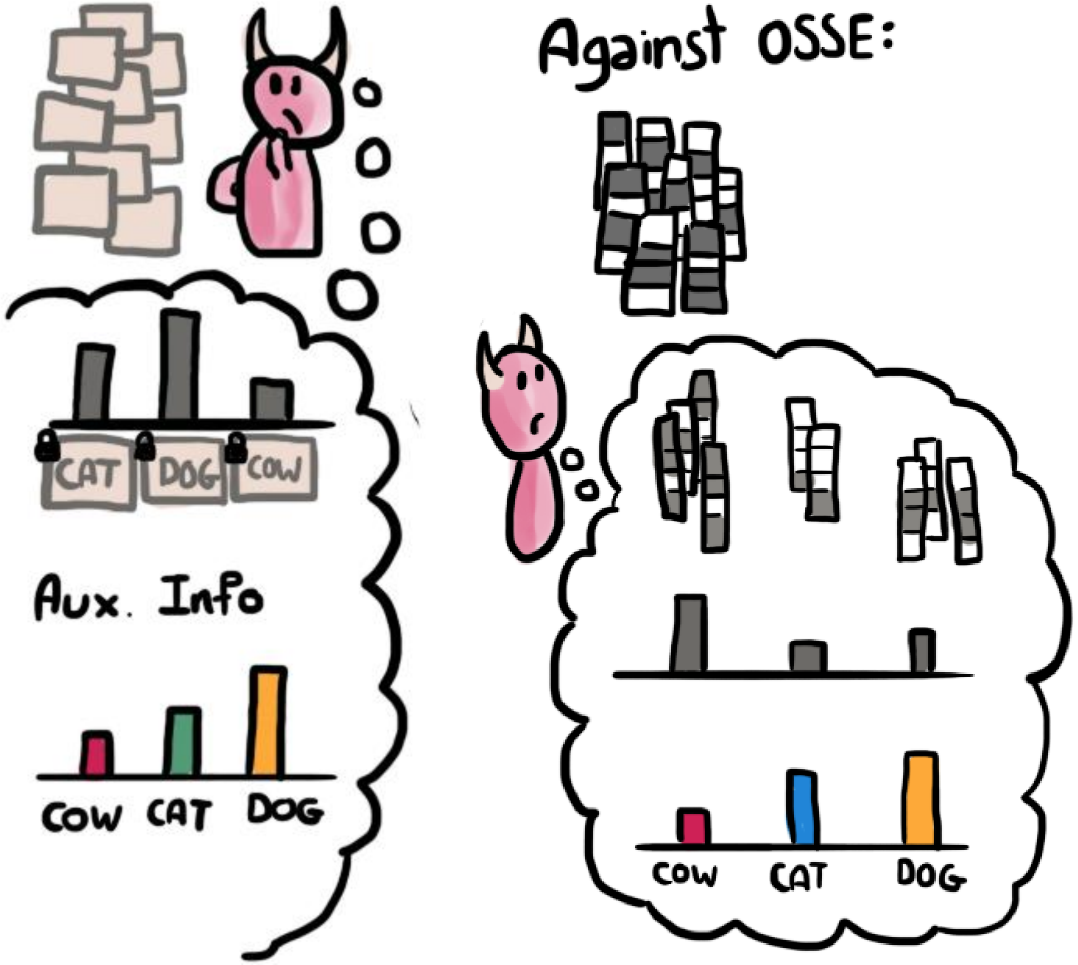
C. Liu, L. Zhu, M. Wang, and Y.-A. Tan, “Search pattern leakage in searchable encryption: Attacks and new construction,” *Information Sciences*, vol. 265, pp. 176–188, 2014.

Evaluation: Frequency Attack



C. Liu, L. Zhu, M. Wang, and Y.-A. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," *Information Sciences*, vol. 265, pp. 176–188, 2014.

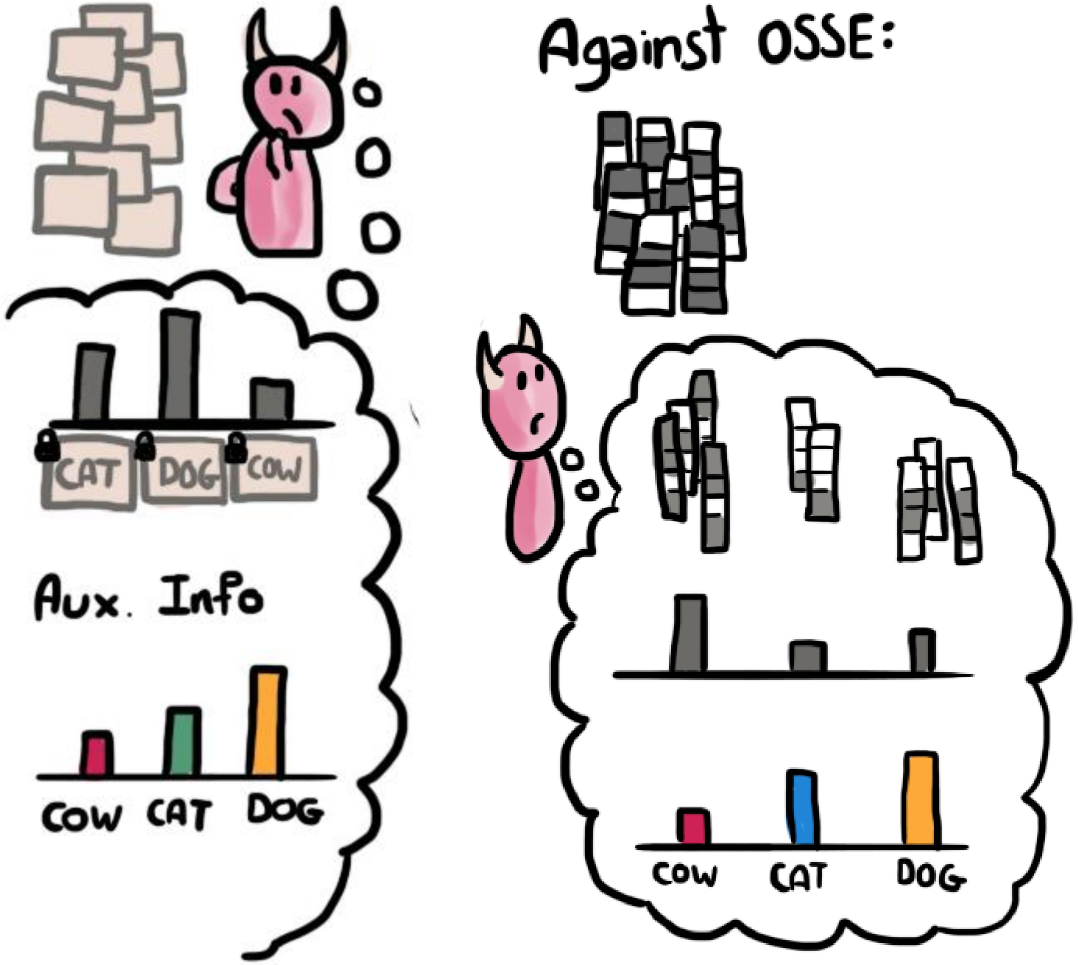
Evaluation: Frequency Attack

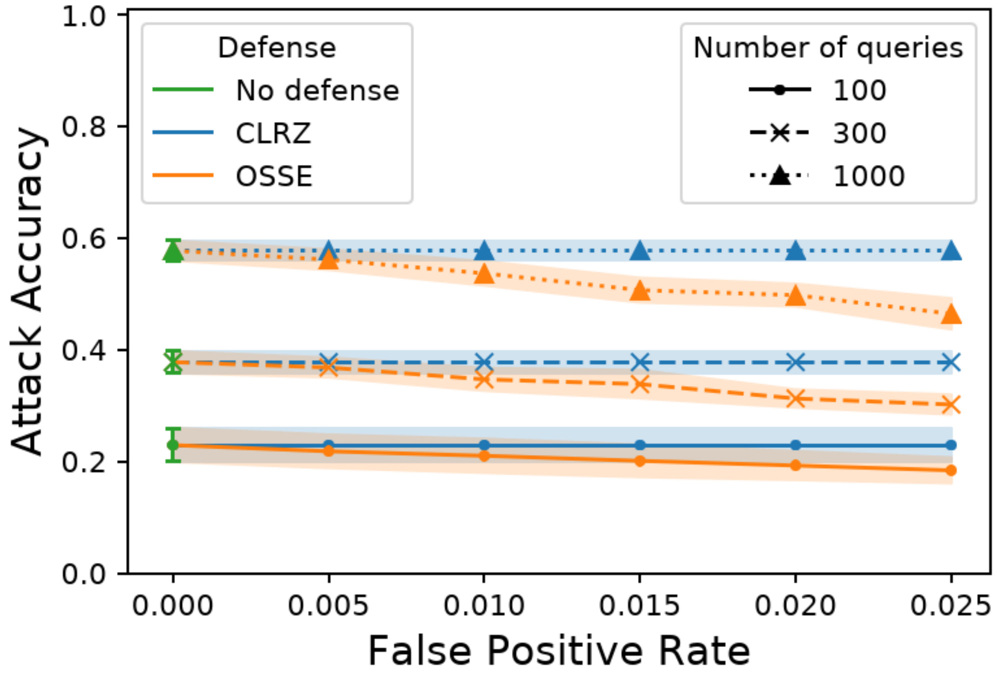


C. Liu, L. Zhu, M. Wang, and Y.-A. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," *Information Sciences*, vol. 265, pp. 176–188, 2014.

Evaluation: Frequency Attack

[redacted]





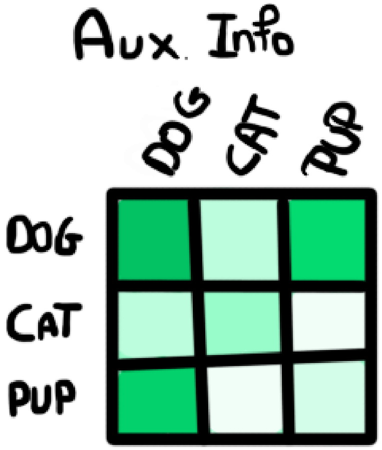
C. Liu, L. Zhu, M. Wang, and Y.-A. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," *Information Sciences*, vol. 265, pp. 176–188, 2014.

Evaluation: IKK

M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.” in *NDSS*, vol. 20, 2012, p. 12.

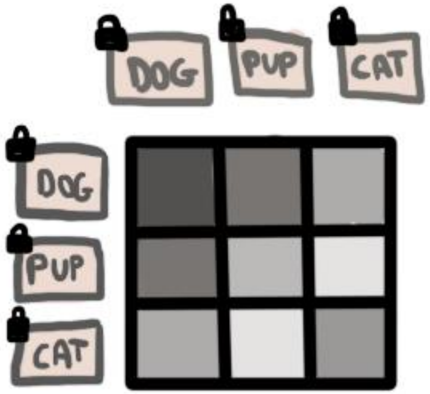
Evaluation: IKK

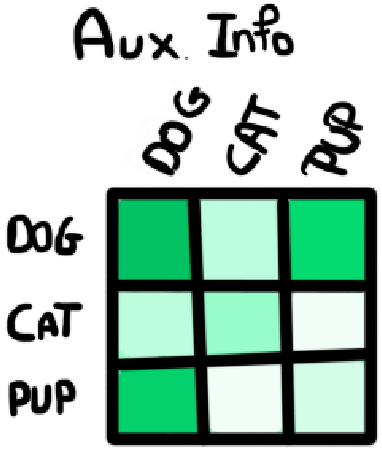
[redacted]



M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.” in *NDSS*, vol. 20, 2012, p. 12.

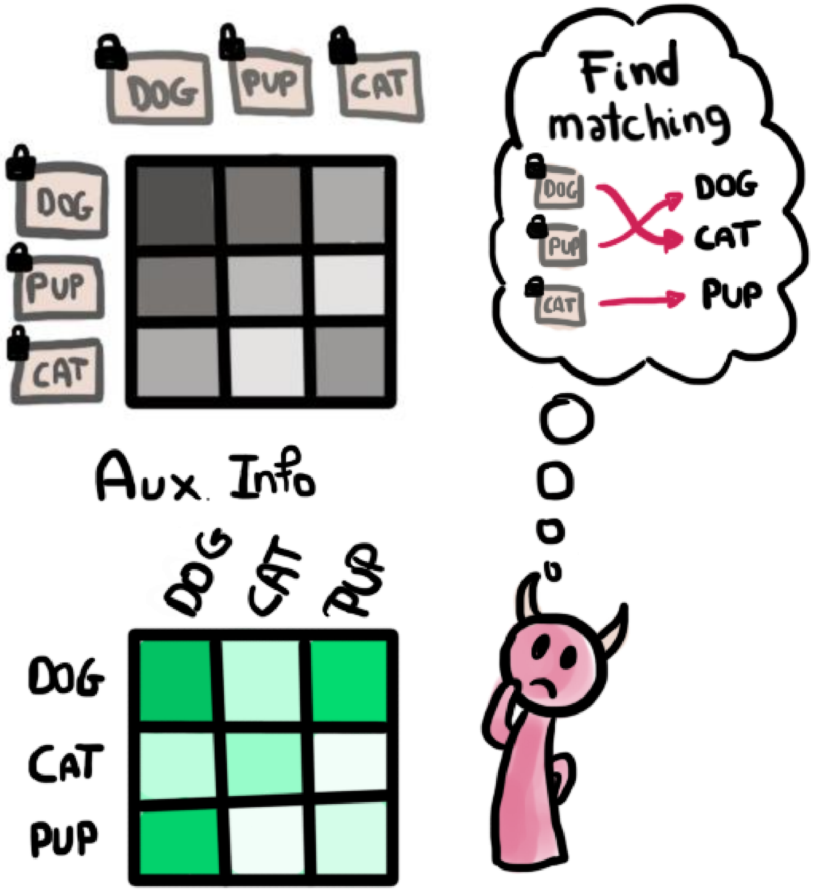
Evaluation: IKK





M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.” in *NDSS*, vol. 20, 2012, p. 12.

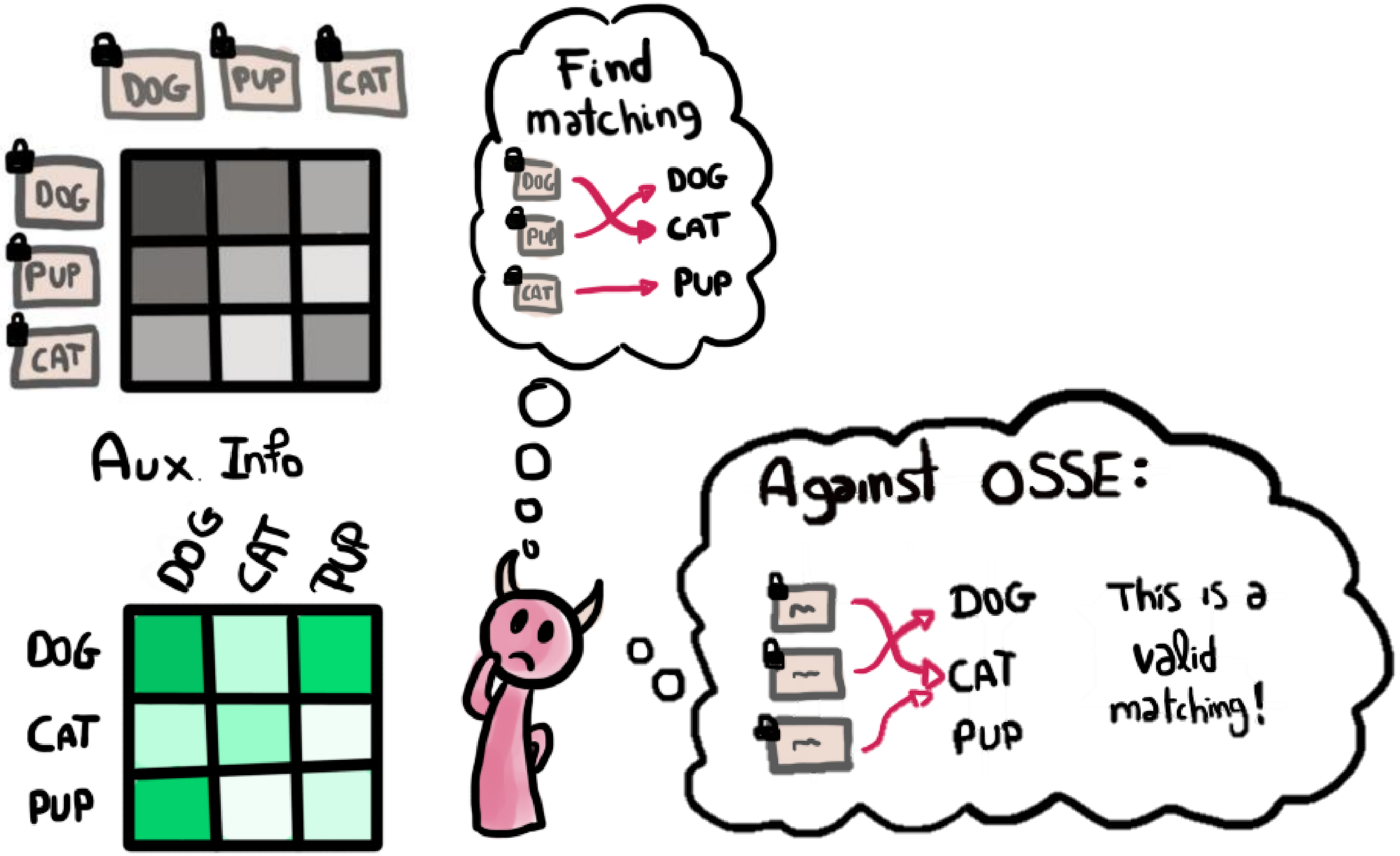
Evaluation: IKK



M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.” in *NDSS*, vol. 20, 2012, p. 12.

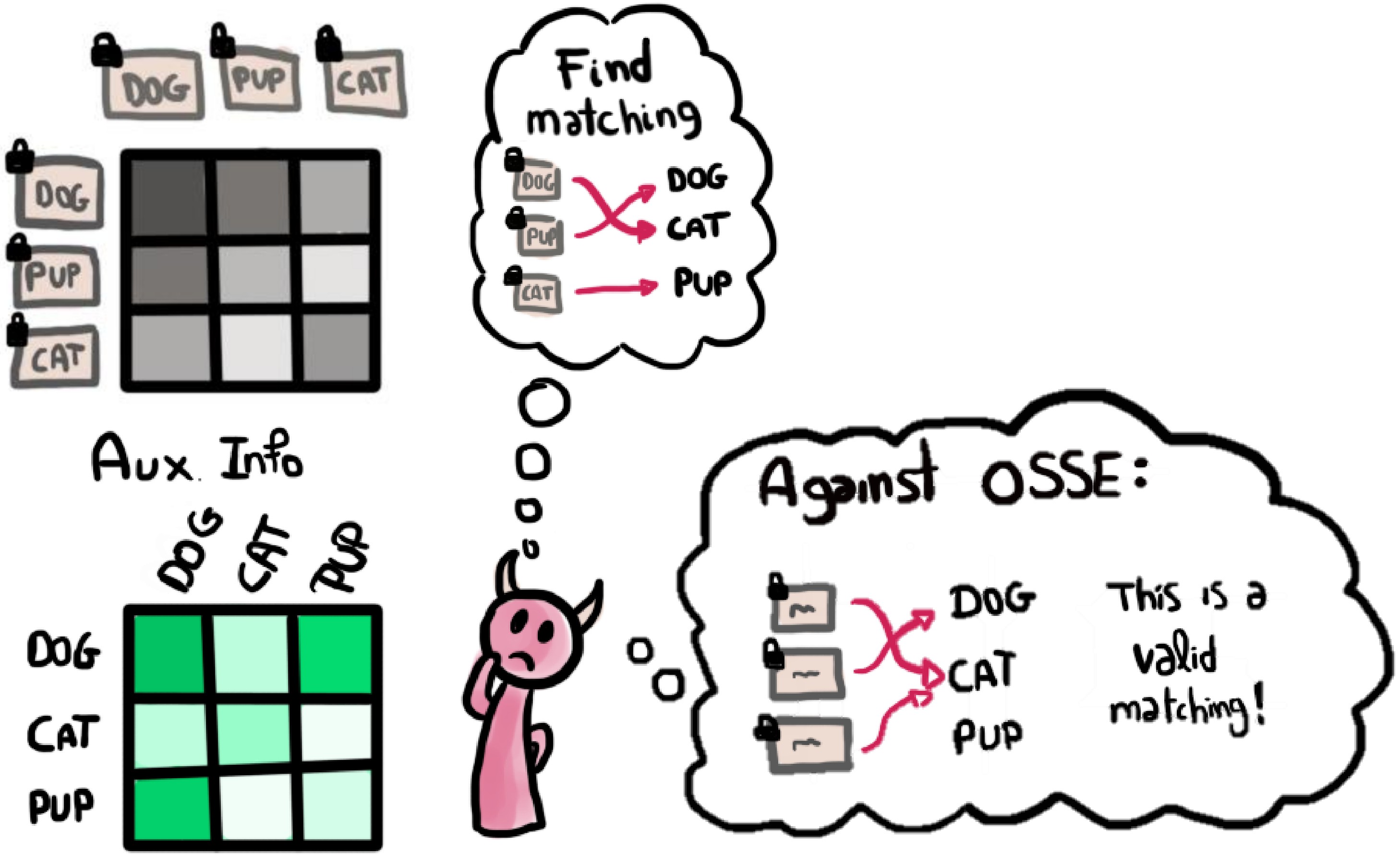
Evaluation: IKK

[redacted]

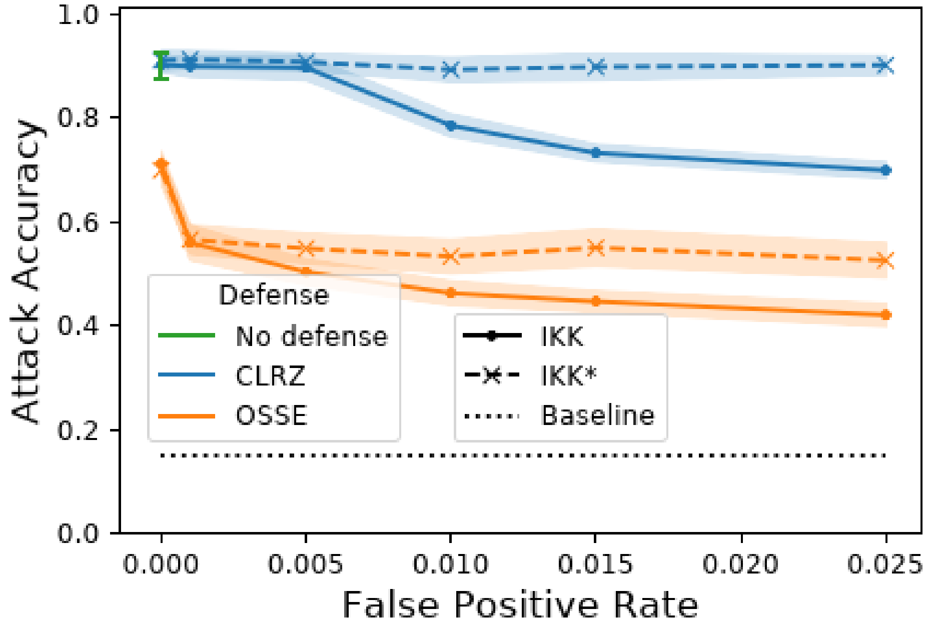


M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation." in *NDSS*, vol. 20, 2012, p. 12.

Evaluation: IKK



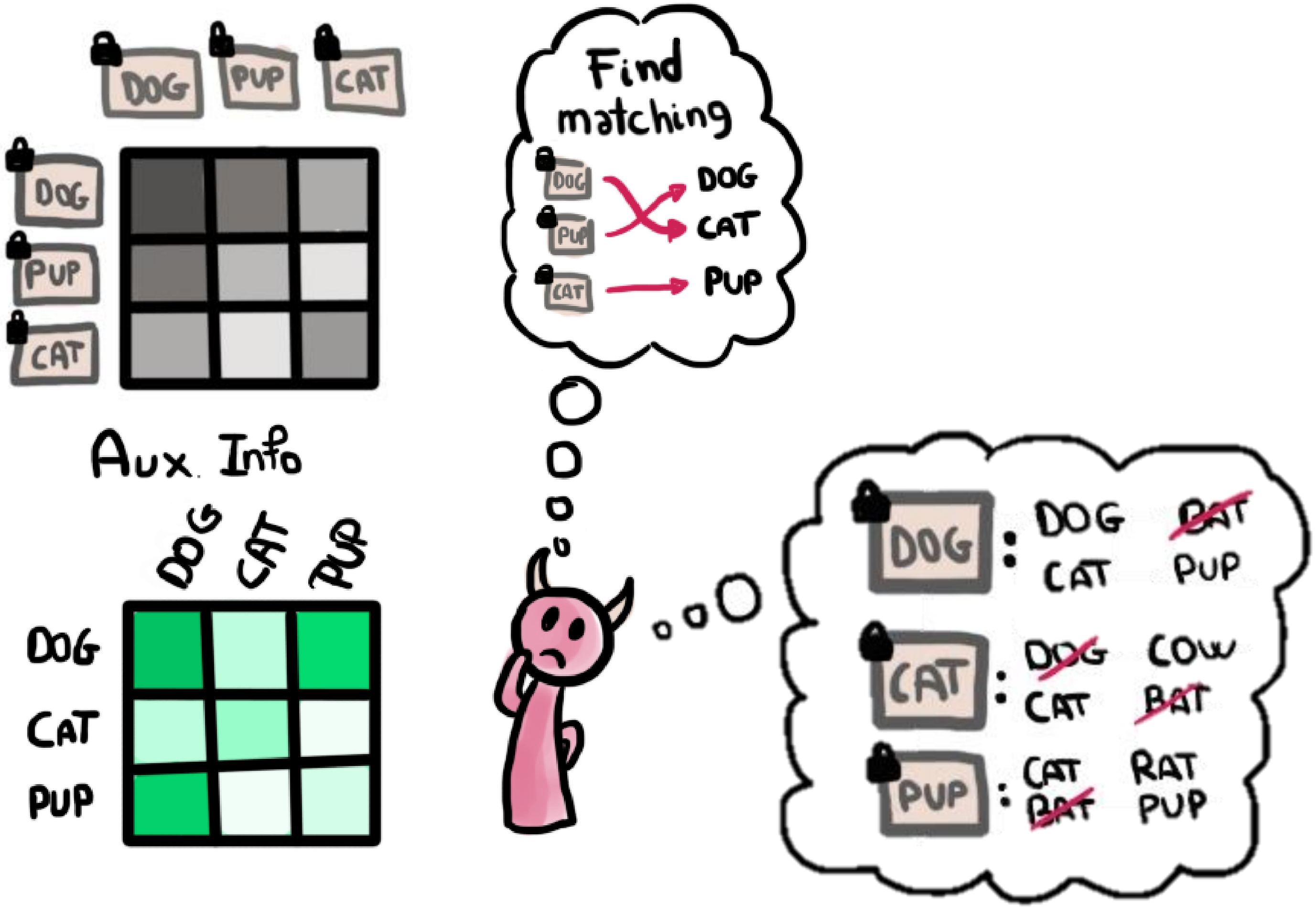
[redacted]



M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation." in *NDSS*, vol. 20, 2012, p. 12.

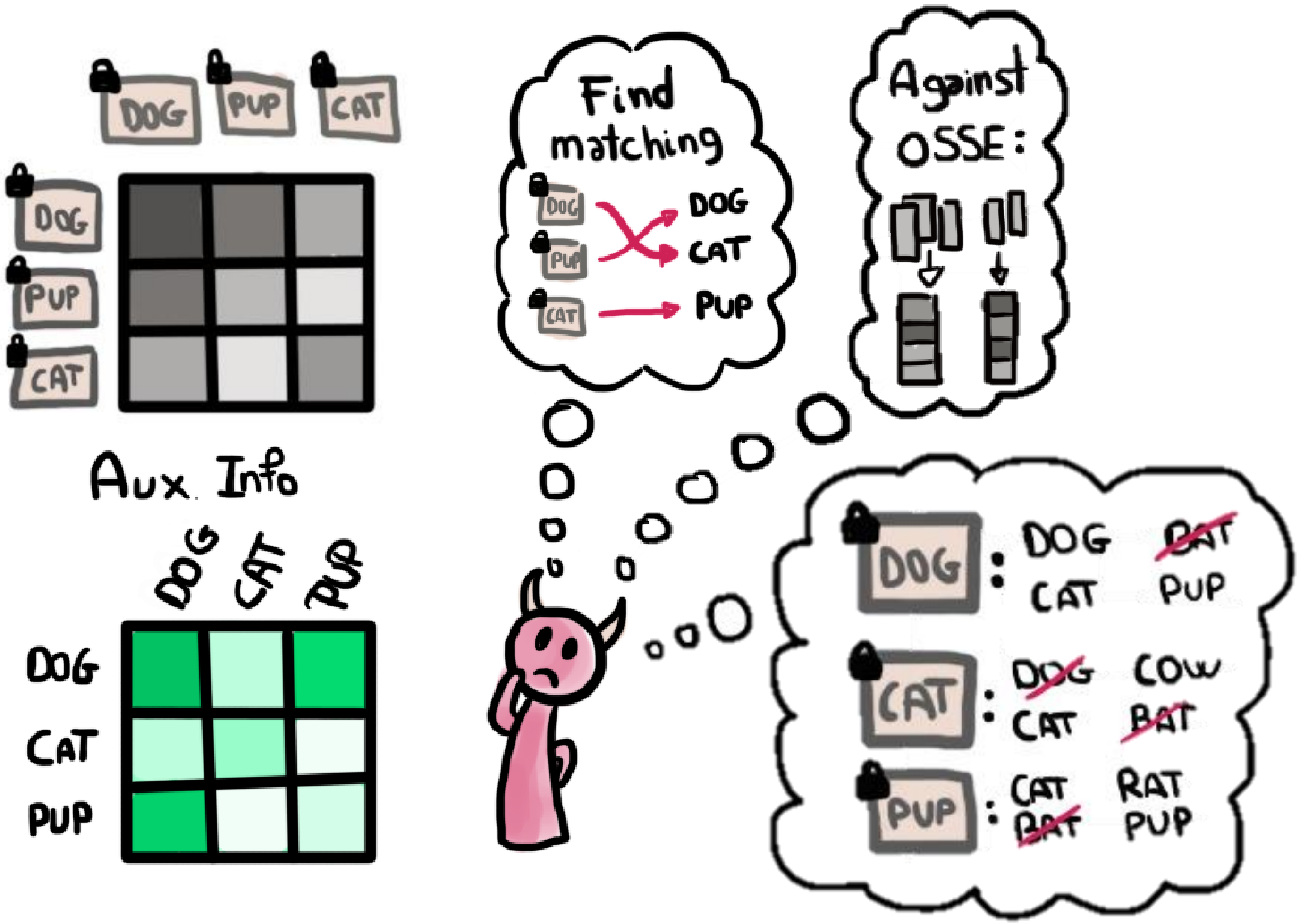
Evaluation: count attack

[redacted]



D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 668–679.

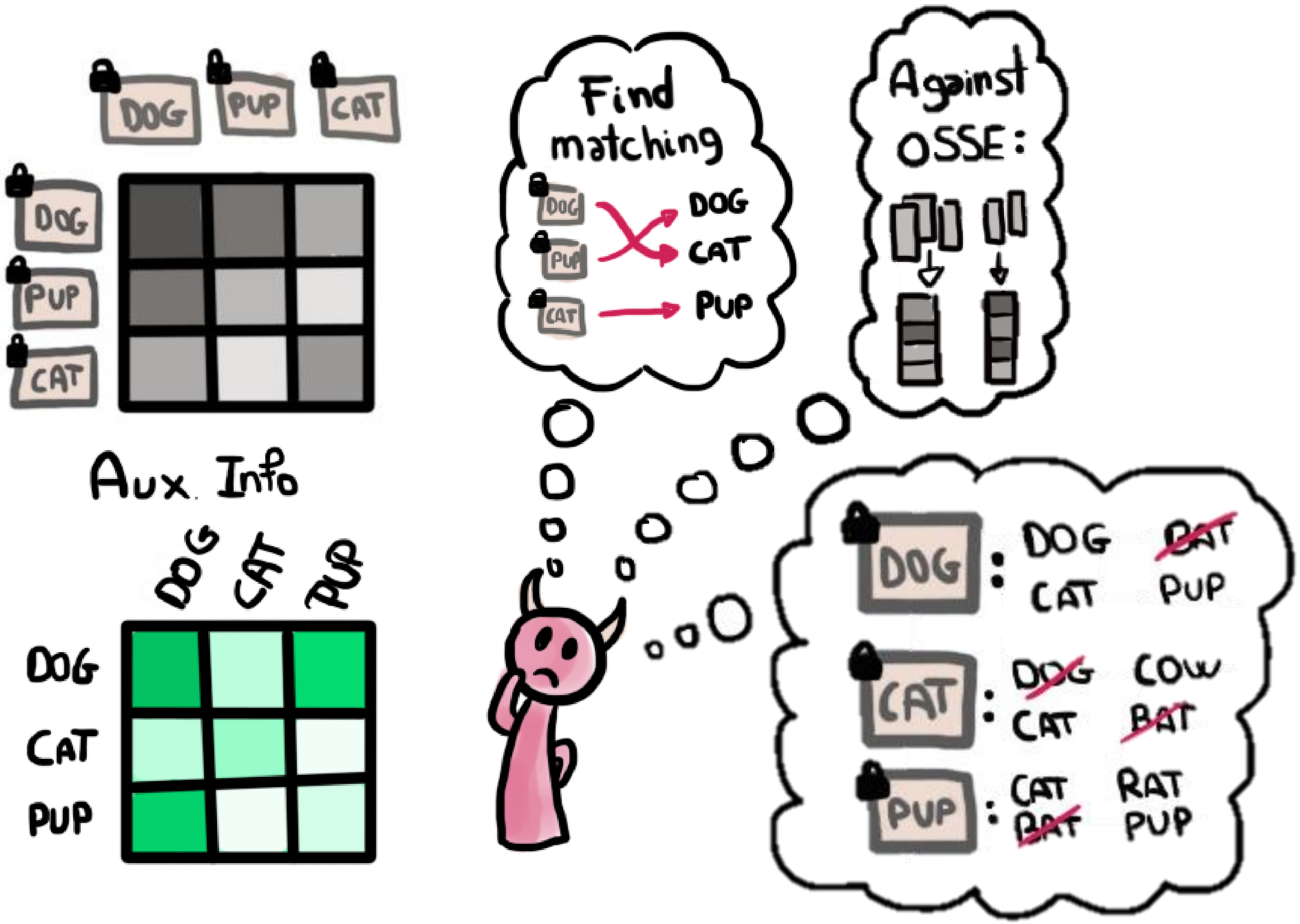
Evaluation: count attack



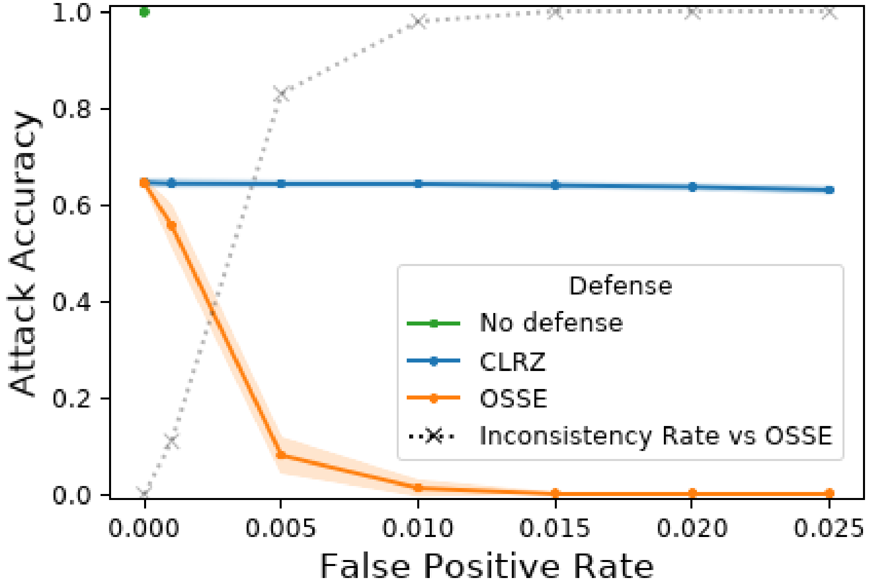
[redacted]

D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 668–679.

Evaluation: count attack

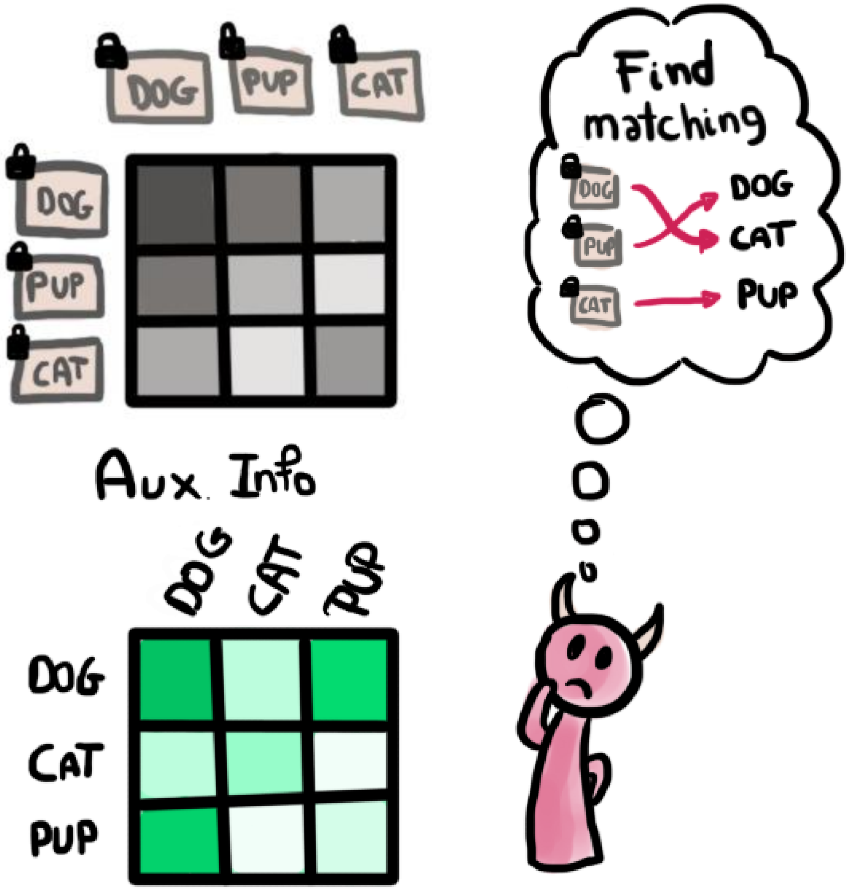


D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 668–679.



Evaluation: graph matching

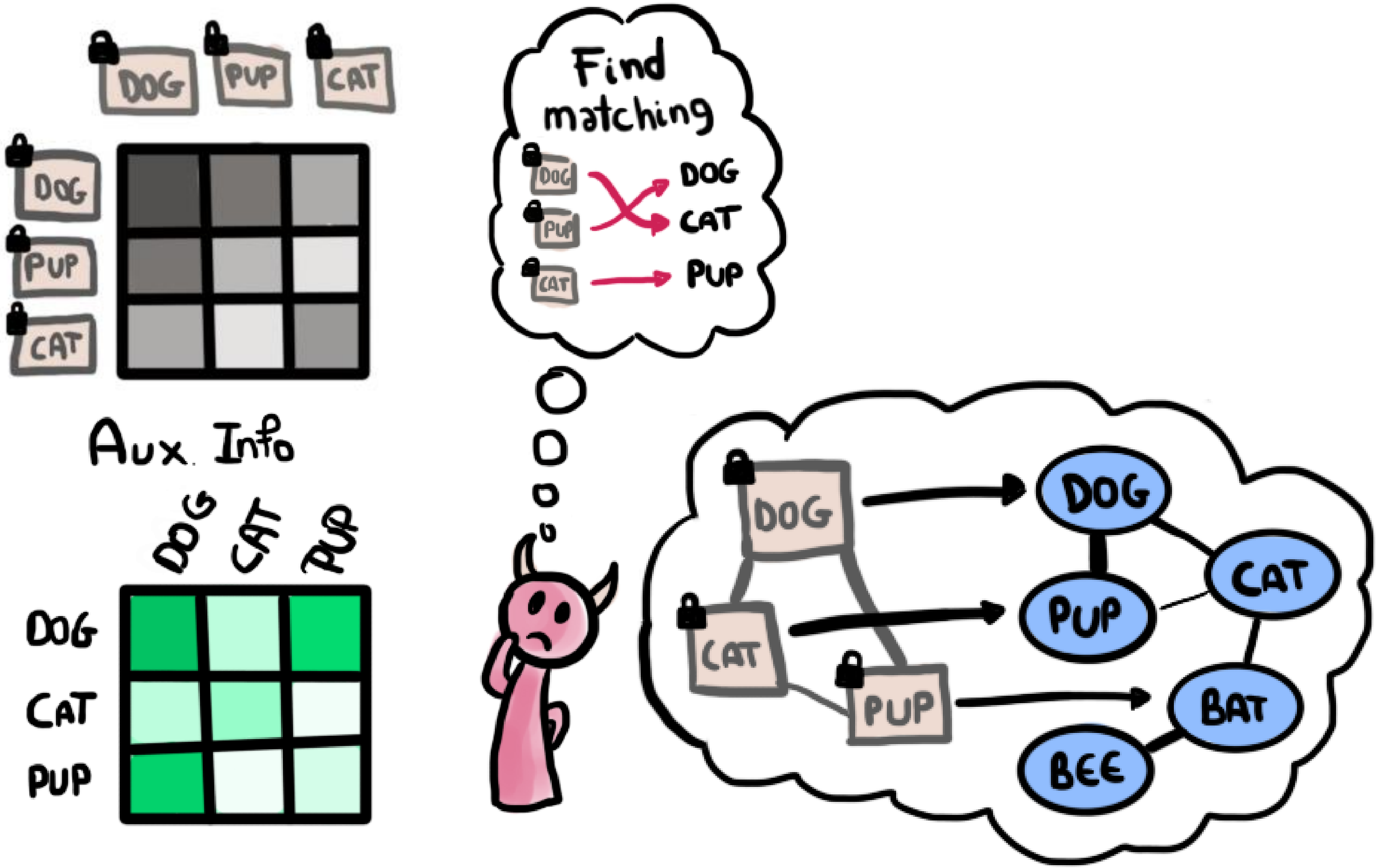
[redacted]



D. Pouliot and C. V. Wright, “The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1341–1352.

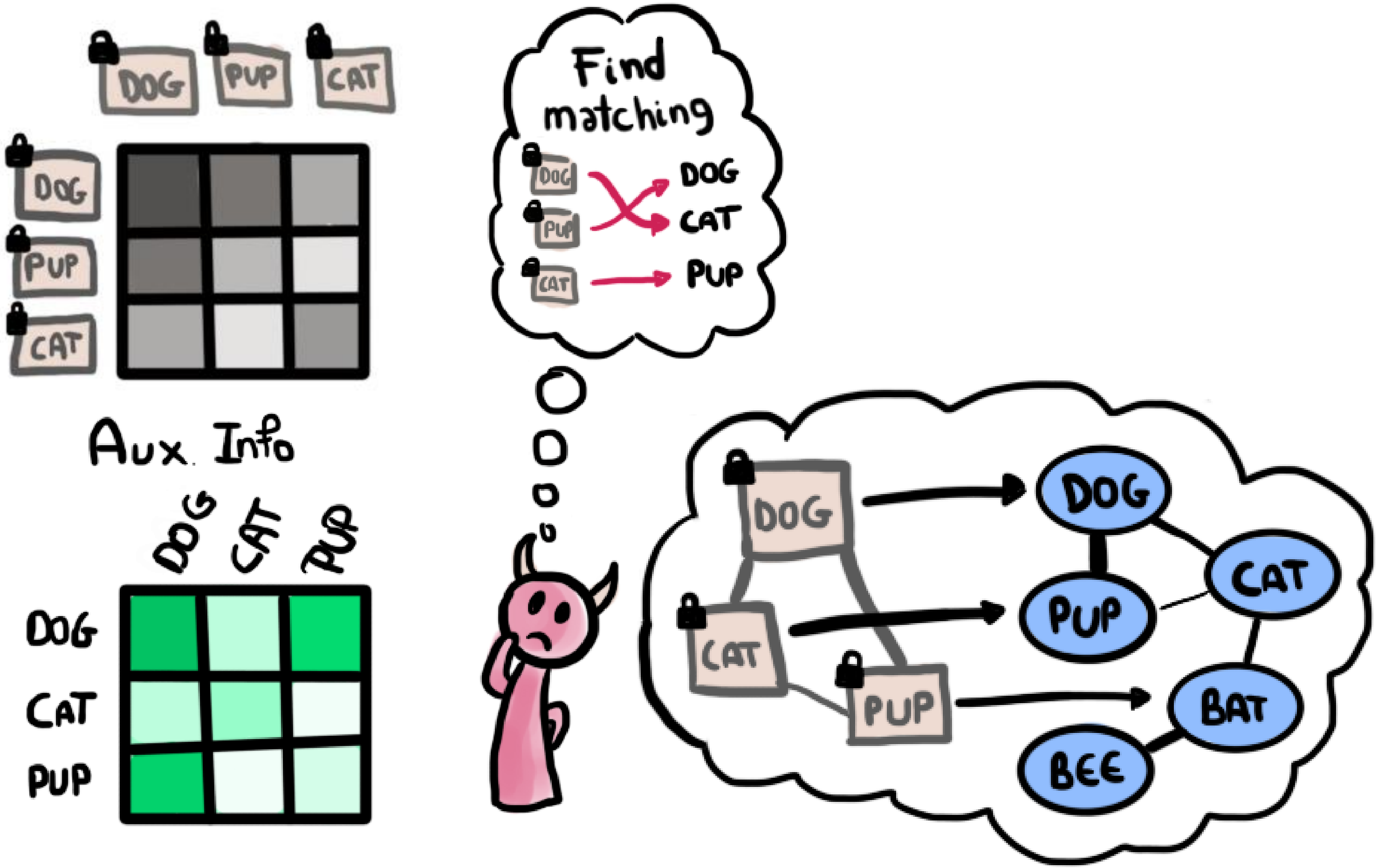
Evaluation: graph matching

[redacted]

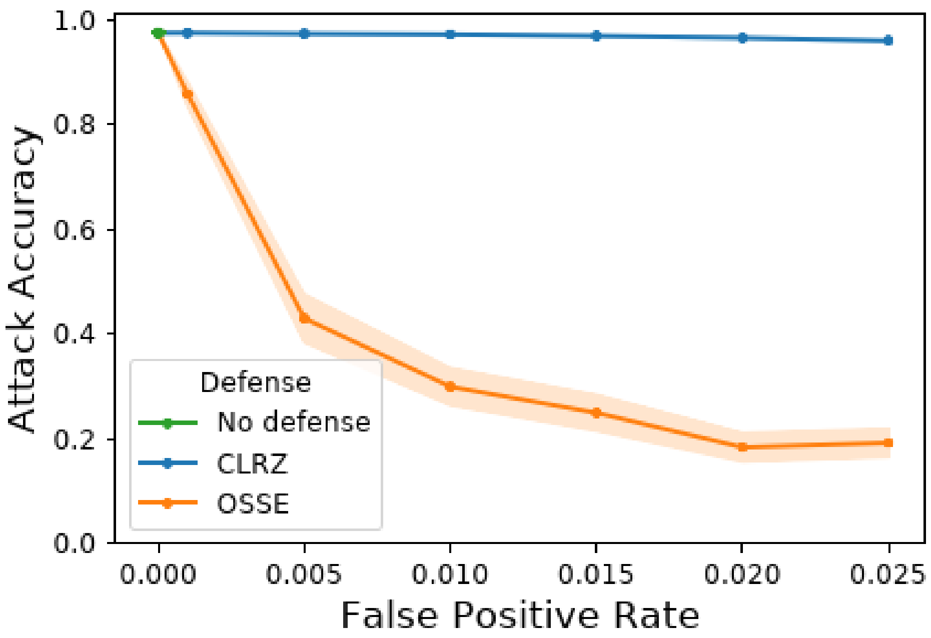


D. Pouliot and C. V. Wright, “The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1341–1352.

Evaluation: graph matching



[redacted]



D. Pouliot and C. V. Wright, "The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1341–1352.

Conclusions

- Hiding search pattern is challenging but very effective against attacks!

Conclusions

- Hiding search pattern is challenging but very effective against attacks!

- OSSE: SSE using IPPE

1 comm round!

No client storage!

Hides search pattern!

Better asymp. comm than ORAM

Conclusions

[redacted]

- Hiding search pattern is challenging but very effective against attacks!

- OSSE: SSE using IPPE

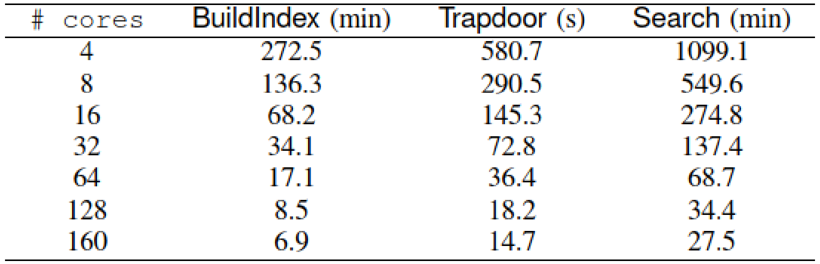
1 comm round!

No client storage!

Better asymp. Comm than ORAM

Hides search pattern!

High computation



# cores	BuildIndex (min)	Trapdoor (s)	Search (min)
4	272.5	580.7	1099.1
8	136.3	290.5	549.6
16	68.2	145.3	274.8
32	34.1	72.8	137.4
64	17.1	36.4	68.7
128	8.5	18.2	34.4
160	6.9	14.7	27.5

TABLE V: Running Times

Conclusions

[redacted]

- Hiding search pattern is challenging but very effective against attacks!

- OSSE: SSE using IPPE

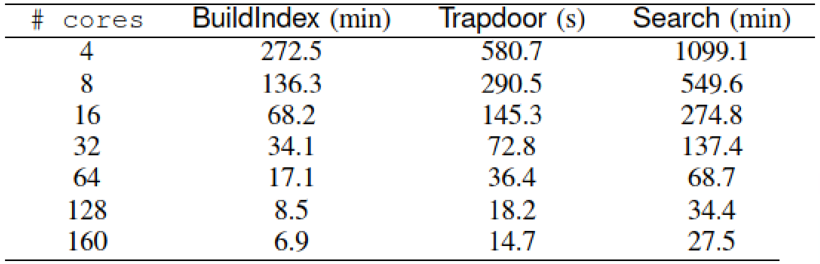
1 comm round!

No client storage!

Hides search pattern!

Better asymp. Comm than ORAM

High computation



# cores	BuildIndex (min)	Trapdoor (s)	Search (min)
4	272.5	580.7	1099.1
8	136.3	290.5	549.6
16	68.2	145.3	274.8
32	34.1	72.8	137.4
64	17.1	36.4	68.7
128	8.5	18.2	34.4
160	6.9	14.7	27.5

TABLE V: Running Times

CLRZ = 200 ms

Conclusions

[redacted]

- Hiding search pattern is challenging but very effective against attacks!

- OSSE: SSE using IPPE

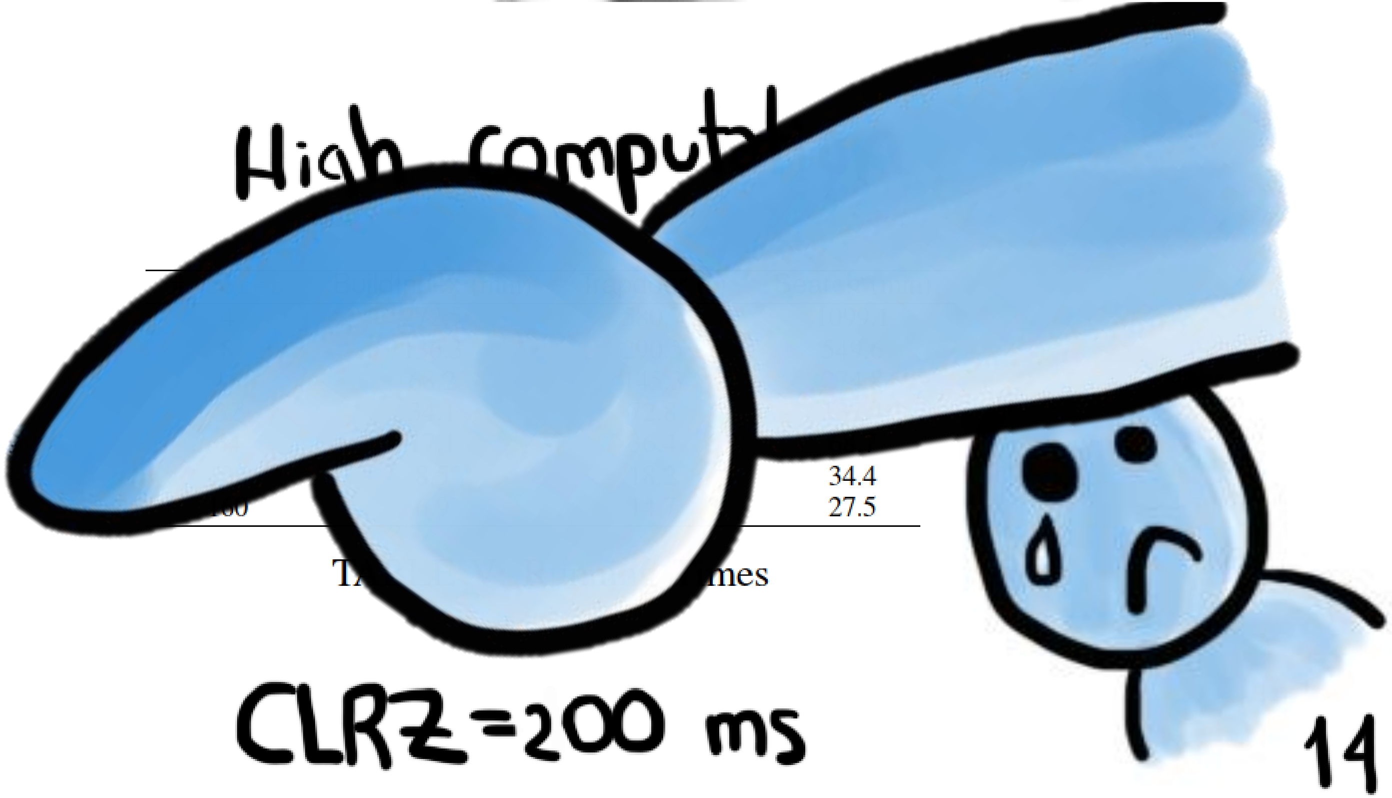
1 comm round!

No client storage!

Hides search pattern!

Better asymp. comm than ORAM

High comput



CLRZ=200 ms

[redacted]

Thanks!

simon.oya@uwaterloo.ca