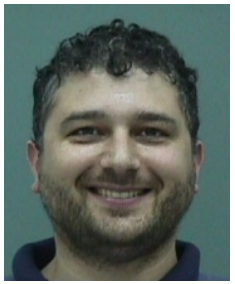
Simultaneous Private Learning of Multiple Concepts

January 16, 2016

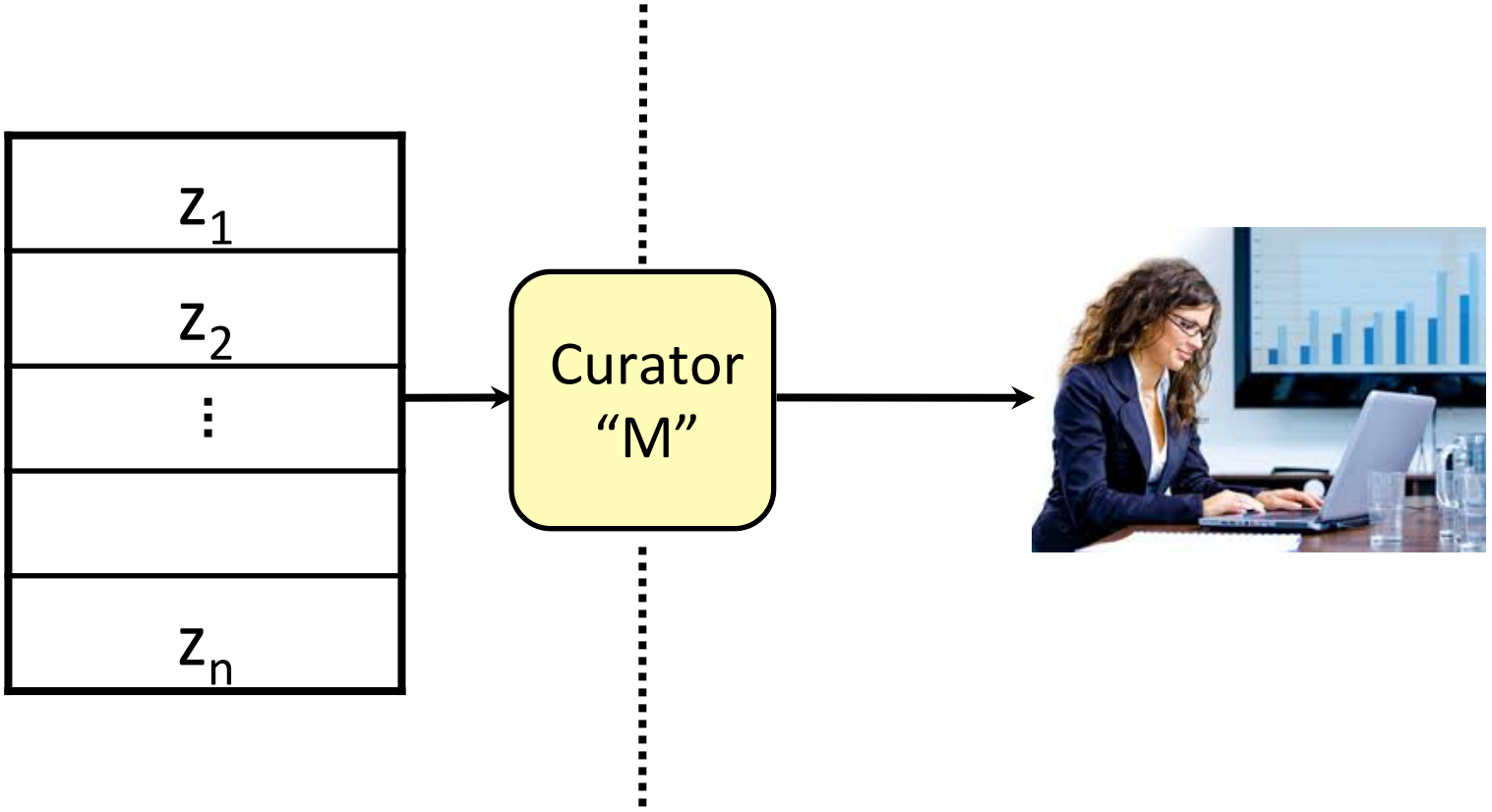


Mark Bun
Harvard U.

Kobbi Nissim
Harvard & Ben-Gurion

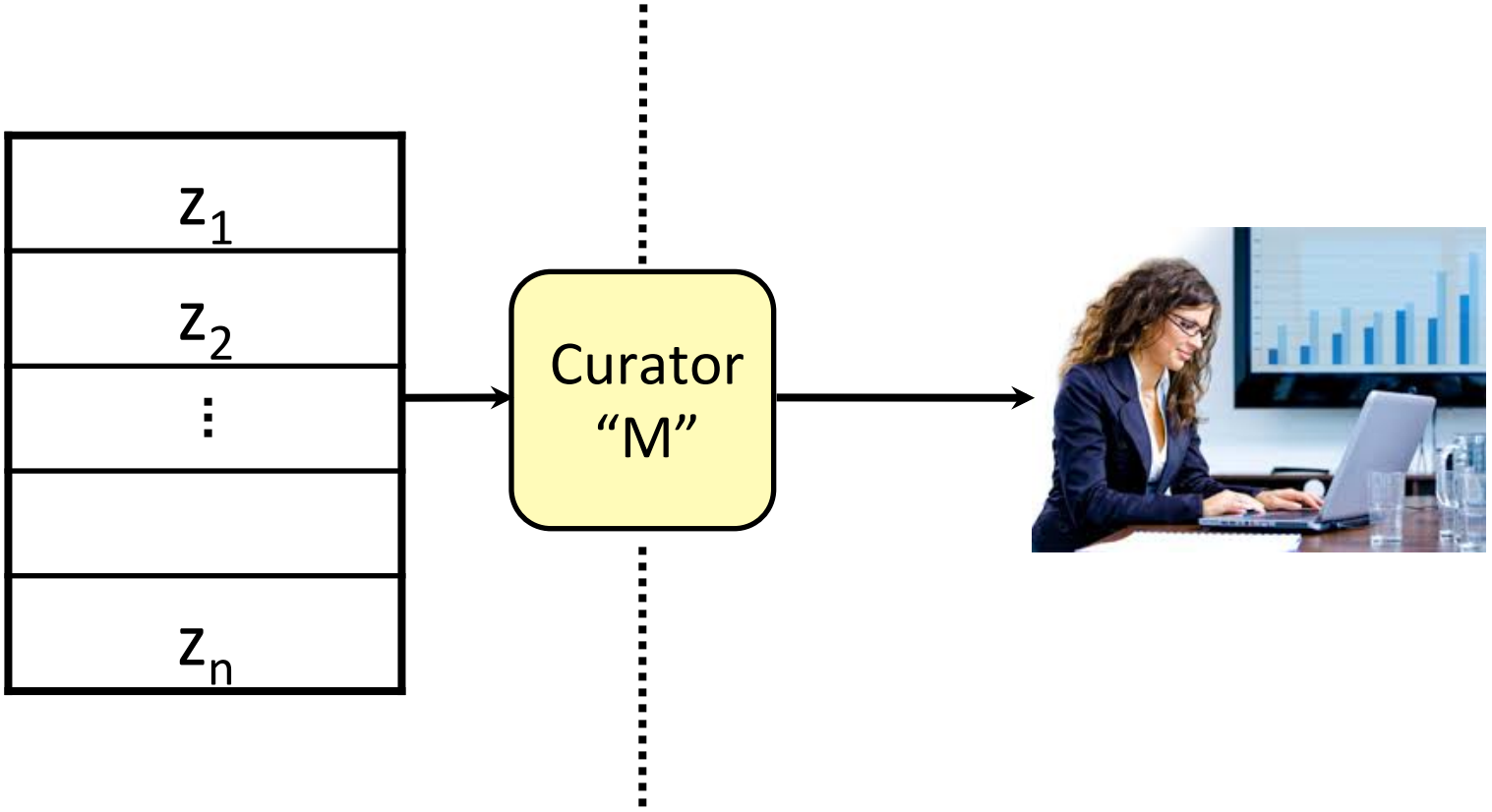
Uri Stemmer
Ben-Gurion U.

Privacy-Preserving Data Analysis



Want curators that are: ◆Private ◆Accurate ◆Efficient

Privacy-Preserving Data Analysis



Want curators that are:

- Differentially Private
- Accurate for Learning Tasks
- Sample Efficient

What can be Done with Differential Privacy?

Histograms [DMNS06]

Contingency tables [BCDKMT07, GHRU11, TUV12, DNT14]

PAC learning [BDMN05, KLNRS08]

Clustering [BDMN05, NRS07]

…and much more!

Streaming algorithms [DNRY10, DNPR10, MMNW11]

SVD [HR12, HR13, KT13, DTTZ14]

Mechanism Design [MT07, NST10, X11, NOS12, CCKMV12, HK12, KPRU12]

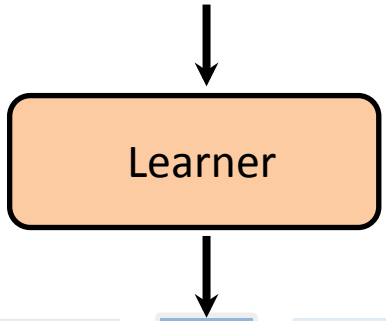
Question: Can these tasks be performed as efficiently as their non-private counterparts?

This work: Sample complexity of privately PAC learning multiple concepts over the same example set

Refs. thanks to Salil Vadhan

PAC Learning [Valiant84]

Gender	Age	4Chan?		MLP:FiM
M	38	Y		1
F	6	N		1
F	34	N		0
…				
M	27	N		0



M	31	Y	h	1

h = ((Age < 10) AND (Gender = F)) OR
((17 < Age < 40) AND (Gender = M) AND (4Chan? = Y))

PAC Learning [Valiant84]

Gender	Age	4Chan?		MLP:FiM
M	38	Y		1
F	6	N		1
F	34	N		0
…				
M	27	N		0

Learner

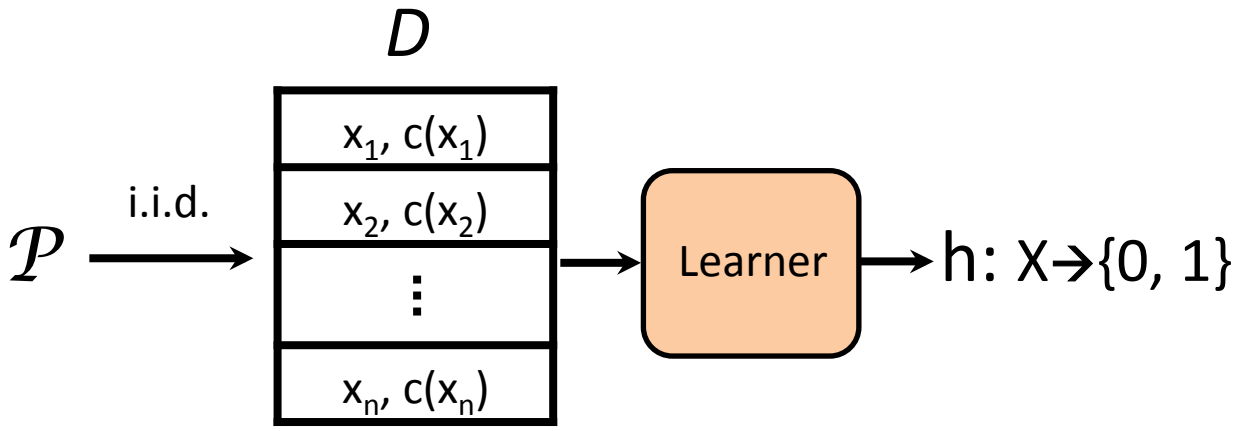
F	65	N	h	0

h = ((Age < 10) AND (Gender = F)) OR
((17 < Age < 40) AND (Gender = M) AND (4Chan? = Y))

PAC Learning [Valiant84]

$\mathcal{P}$ = unknown distribution over domain X

$\mathcal{C}$ = concept class {c: X→{0, 1}} e.g. DNF of intervals

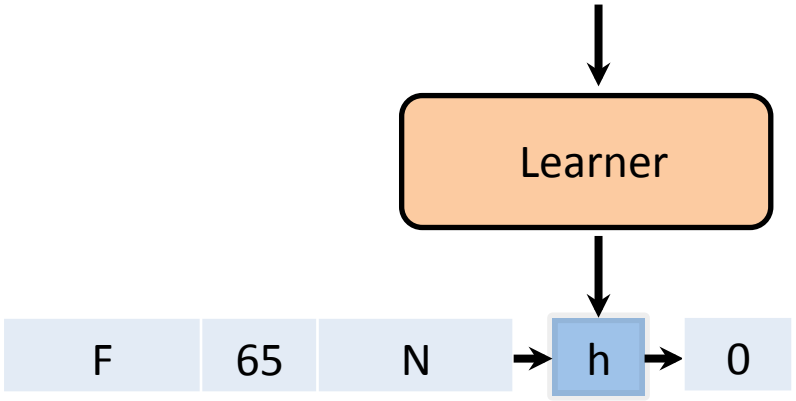


Fact: ***n*** = Θ(VC($\mathcal{C}$)) samples suffice to generalize

VC($\mathcal{C}$) ≤ log|$\mathcal{C}$|, but can be much smaller

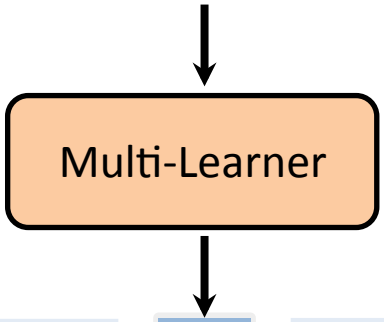
PAC *Multi*-Learning [Valiant06]

Gender	Age	4Chan?		MLP:FiM
M	38	Y		1
F	6	N		1
F	34	N		0
…				
M	27	N		0



PAC *Multi*-Learning [Valiant06]

Gender	Age	4Chan?		MLP:FiM	AdvTime	A:tLA	Dora
M	38	Y		1	1	1	0
F	6	N		1	0	0	0
F	34	N		0	0	1	0
…							
M	27	N		0	1	0	0

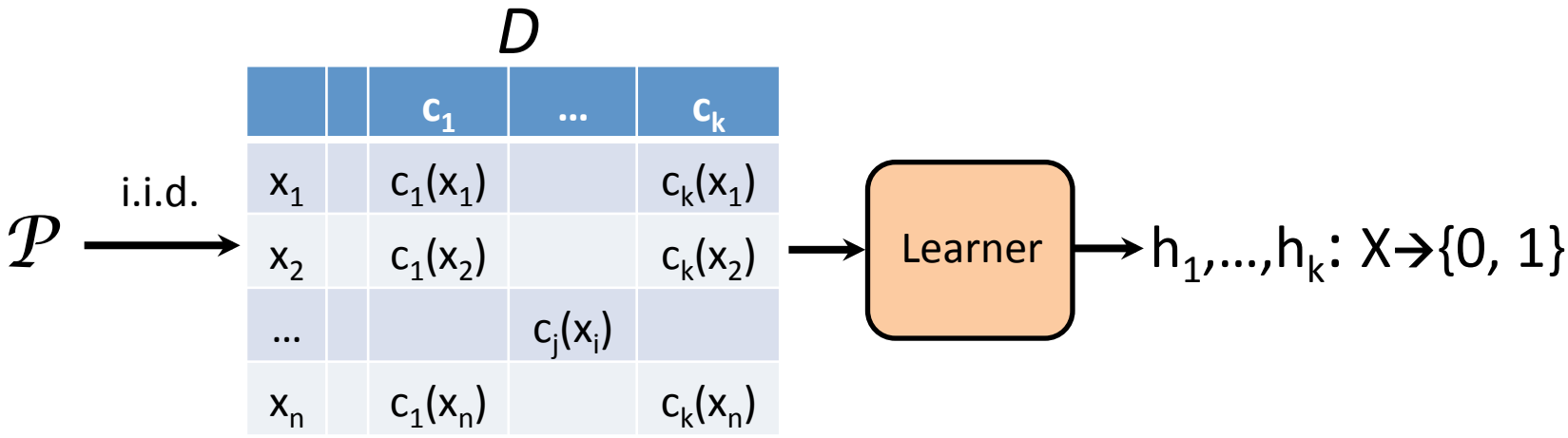


F	65	N	→ **h** →	0	0	1	0

PAC *Multi*-Learning [Valiant06]

$\mathcal{P}$ = unknown distribution over domain X

$\mathcal{C}$ = concept class {c: X→{0, 1}}



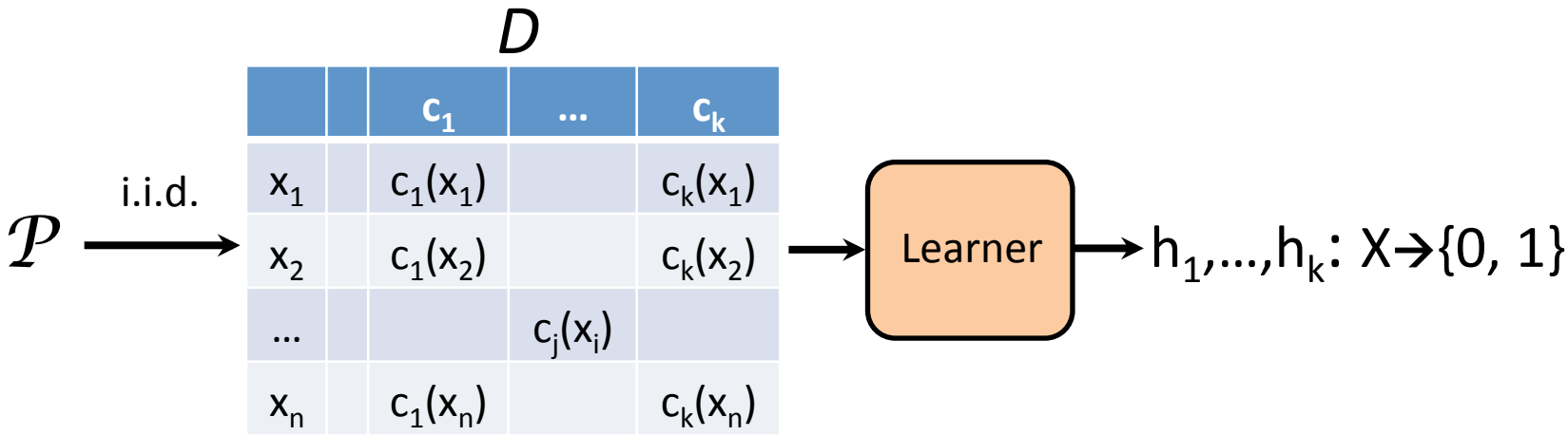
Goal: For all $\mathcal{P}$ and $c_1,\ldots,c_k \in \mathcal{C}$, output **h** s.t.

$h_i \approx c_i$ on $\mathcal{P}$ for every i=1,…,k

PAC *Multi*-Learning [Valiant06]

$\mathcal{P}$ = unknown distribution over domain X

$\mathcal{C}$ = concept class {c: X→{0, 1}}



Fact: ***n*** = Θ(VC($\mathcal{C}$)) samples suffice to generalize

Uniform convergence: Over a random sample S of size $O_{\alpha,\beta}(VC(\mathcal{C}))$,

$$\Pr\left[\exists f,g \in C : (f|_S = g|_S) \wedge err_P(f,g) > \alpha\right] \leq \beta$$

What about Privacy?

Gender	Age	4Chan?		MLP:FiM	AdvTime	A:tLA	Dora
M	38	Y		1	1	1	0
F	6	N		1	0	0	0
F	34	N		0	0	1	0
…							
M	27	N		0	1	0	0

The data is anonymized, so it's safe to release, right?

F	65	N	→	*h*	→	0	0	1	0

What about Privacy?

Gender	Age	4Chan?		MLP:FiM
M	38	Y		1
F	6	N		1
F	34	N		0
…				
M	27	N		0

The data is anonymized, so it's safe to release, right?

F	65	N	→ ***h*** →	0

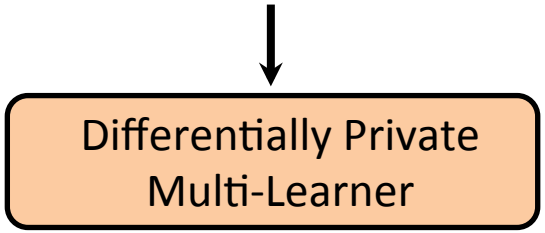
Wrong! [Narayanan-Shmatikov08]

Motivates need for rigorous privacy guarantees

Private PAC Multi-Learning

Extending Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith '08

Attributes		c_1	c_2	…	c_k
x_1		$c_1(x_1)$	$c_2(x_1)$		$c_k(x_1)$
x_2		$c_1(x_2)$	$c_2(x_2)$		$c_k(x_2)$
x_3		$c_1(x_3)$	$c_2(x_3)$		$c_k(x_3)$
…				$c_j(x_i)$	
x_n		$c_1(x_n)$	$c_2(x_n)$		$c_k(x_n)$

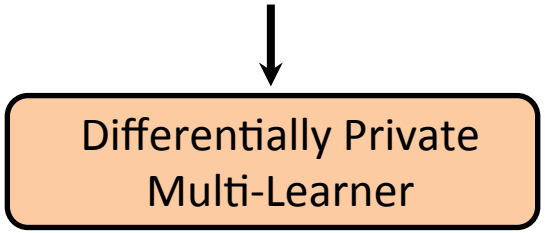


h_1	h_2	…	h_k

Private PAC Multi-Learning

Extending Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith '08

Attributes		c_1	c_2	…	c_k
x_1		b_{11}	b_{21}		b_{k1}
x_2		b_{12}	b_{22}		b_{k2}
x_3		b_{13}	b_{23}		b_{k3}
…				b_{ji}	
x_n		b_{1n}	b_{2n}		b_{kn}

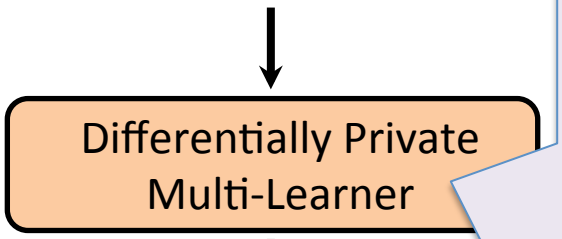


h_1	h_2	…	h_k

Private PAC Multi-Learning

Extending Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith '08

Attributes		c_1	c_2	…	c_k
x_1		b_{11}	b_{21}		b_{k1}
x_2		b_{12}	b_{22}		b_{k2}
x_3		b_{13}	b_{23}		b_{k3}
…				b_{ji}	
x_n		b_{1n}	b_{2n}		b_{kn}



h_1	h_2	…	h_k

D and *D'* are **neighbors** if they differ on one row

M is **differentially private** if for all neighbors *D*, *D'* :

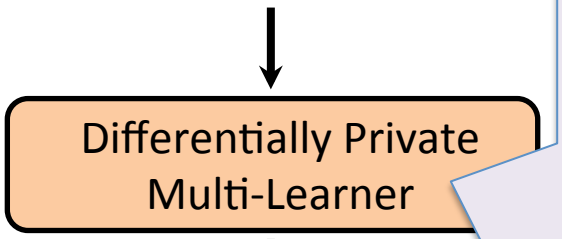
$$M(D) \approx M(D')$$

DN03+Dwork, DN04, BDMN05, **DMNS06**, **DKMMN06**

Private PAC Multi-Learning

Extending Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith '08

Attributes		c_1	c_2	…	c_k
x_1		b_{11}	b_{21}		b_{k1}
x_2		b_{12}	b_{22}		b_{k2}
x'_3		b'_{13}	b'_{23}		b'_{k3}
…				b_{ji}	
x_n		b_{1n}	b_{2n}		b_{kn}



h_1	h_2	…	h_k

D and *D'* are **neighbors** if they differ on one row

M is **differentially private** if for all neighbors *D*, *D'* :

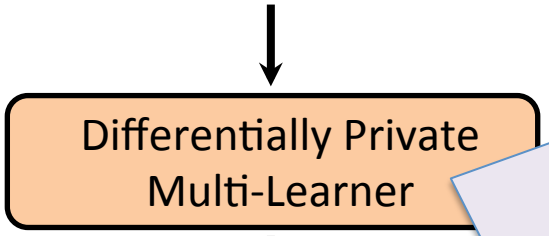
$$M(D) \approx M(D')$$

DN03+Dwork, DN04, BDMN05, **DMNS06**, **DKMMN06**

Private PAC Multi-Learning

Extending Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith '08

Attributes		c_1	c_2	…	c_k
x_1		b_{11}	b_{21}		b_{k1}
x_2		b_{12}	b_{22}		b_{k2}
x'_3		b'_{13}	b'_{23}		b'_{k3}
…				b_{ji}	
x_n		b_{1n}	b_{2n}		b_{kn}



h_1	h_2	…	h_k

D and *D'* are **neighbors** if they differ on one row

M is **(ε,δ)-differentially private** if for all neighbors *D*, *D'* and T⊆Range(M):

$$\Pr[M(D') \in T] \le (1+\varepsilon)\Pr[M(D) \in T] + \delta$$

DN03+Dwork, DN04, BDMN05, **DMNS06**, **DKMMN06**

Question: How does the sample complexity depend on *k*?

Samp. Cx. of Private Multi-Learning

- For $k = 1$, can privately learn $\mathcal{C}$ with sample complexity $n = \mathrm{SCDP}_1(\mathcal{C})$ where:

$$\mathrm{VC}(\mathcal{C}) \le \mathrm{SCDP}_1(\mathcal{C}) \le \log |\mathcal{C}| \quad \text{[KLNRS08]}$$

- For arbitrary k, can learn each concept independently: $\mathrm{SCDP}_k(\mathcal{C}) \le k^{1/2}\, \mathrm{SCDP}_1(\mathcal{C})$ [DRV10]

- Can we do better? Is the dependence on k necessary?

Our Results

Upper bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)
$\texttt{POINT}_X$	1	$\sqrt{k}$
$\texttt{THRESH}_X$	$2^{\log^* \lvert X\rvert} + \sqrt{k}$	
General C	$\min\{\sqrt{k}\log\lvert C\rvert, \sqrt{k}\,\mathrm{VC}(C) + \log\lvert X\rvert\,\mathrm{VC}(C), \sqrt{k}\,\mathrm{VC}(C) + \sqrt{\log\lvert X\rvert}\log\lvert C\rvert\}$	
$\texttt{PAR}_d$ (uniform)	$\log\lvert C\rvert$	$\sqrt{k}\log\lvert C\rvert$

Lower bounds:

C	PAC learning		Agnostic learning	
	proper	improper	proper	improper
$\texttt{POINT}_X$	1		$\sqrt{k}$	
$\texttt{THRESH}_X$	$\log^* \lvert X\rvert + k^{1/3}$	$k^{1/3}$	$\log^* \lvert X\rvert + \sqrt{k}$	$\sqrt{k}$
$\texttt{PAR}_d$ (uniform)	$\log\lvert C\rvert$		$\sqrt{k} + \log\lvert C\rvert$	

Our Results (Human Readable Version)

- Upper bounds
  - Generic multi-learner achieving

Fixed cost

Marginal cost $\leq \mathrm{SCDP}_1(C)$

$$\mathrm{SCDP}_k(C) \leq \mathrm{VC}(C) \log |X| + k^{1/2} \mathrm{VC}(C)$$

  - Improved multi-learners for specific classes

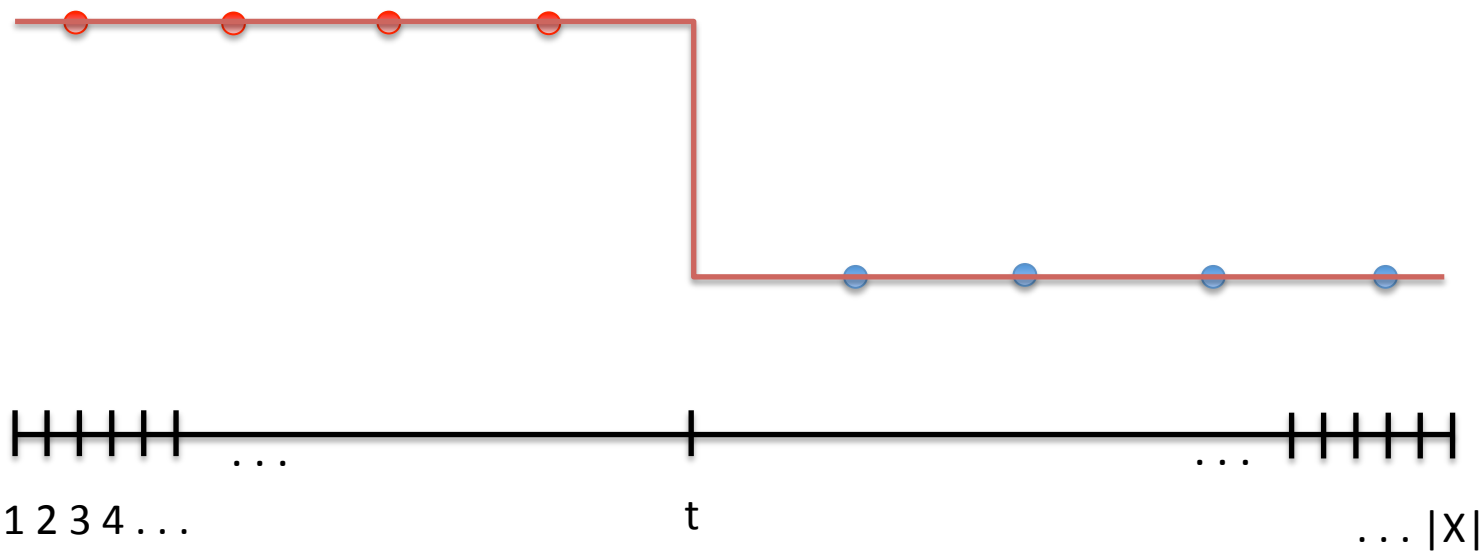
- Lower bounds via fingerprinting codes
  - $k^{1/3}$ lower bound for multi-learning thresholds
  - $k^{1/2}$ lower bound for agnostically learning…

    …anything

Threshold Functions

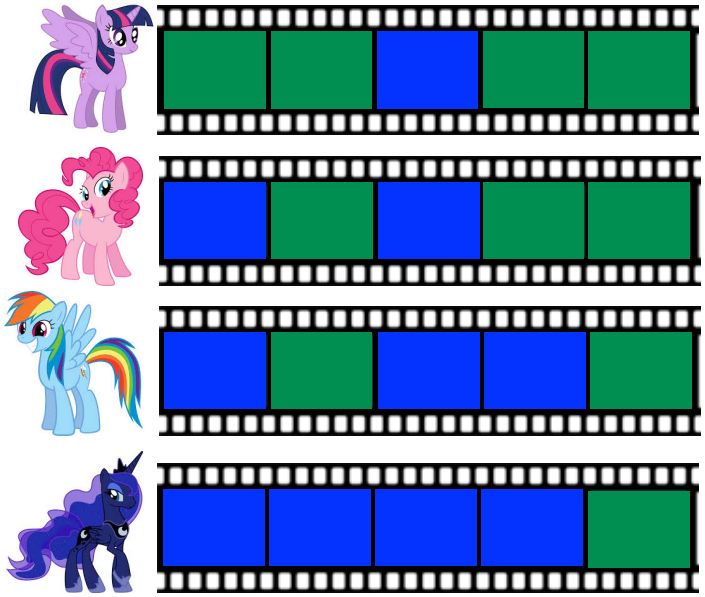
X a totally ordered domain

$\mathcal{C} = \{f_t : f_t(x) = 1 \text{ iff } x \leq t\}$

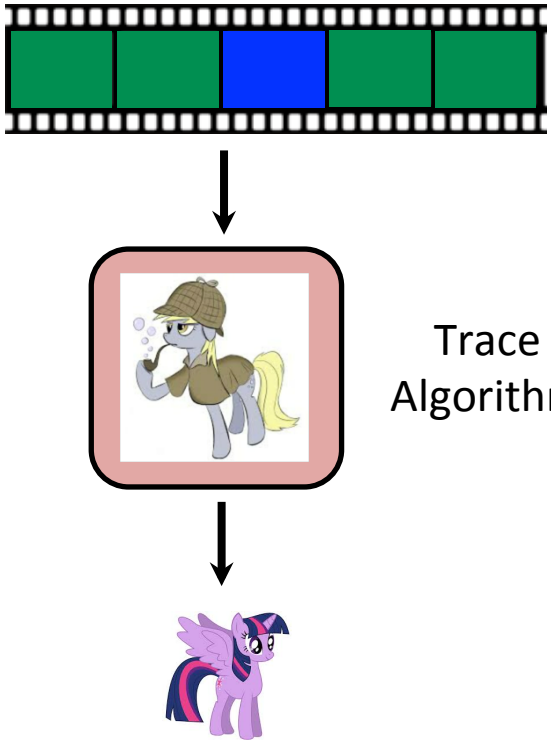


Fingerprinting Codes [Boneh-Shaw95]

I want to distribute my new movie



Pirate

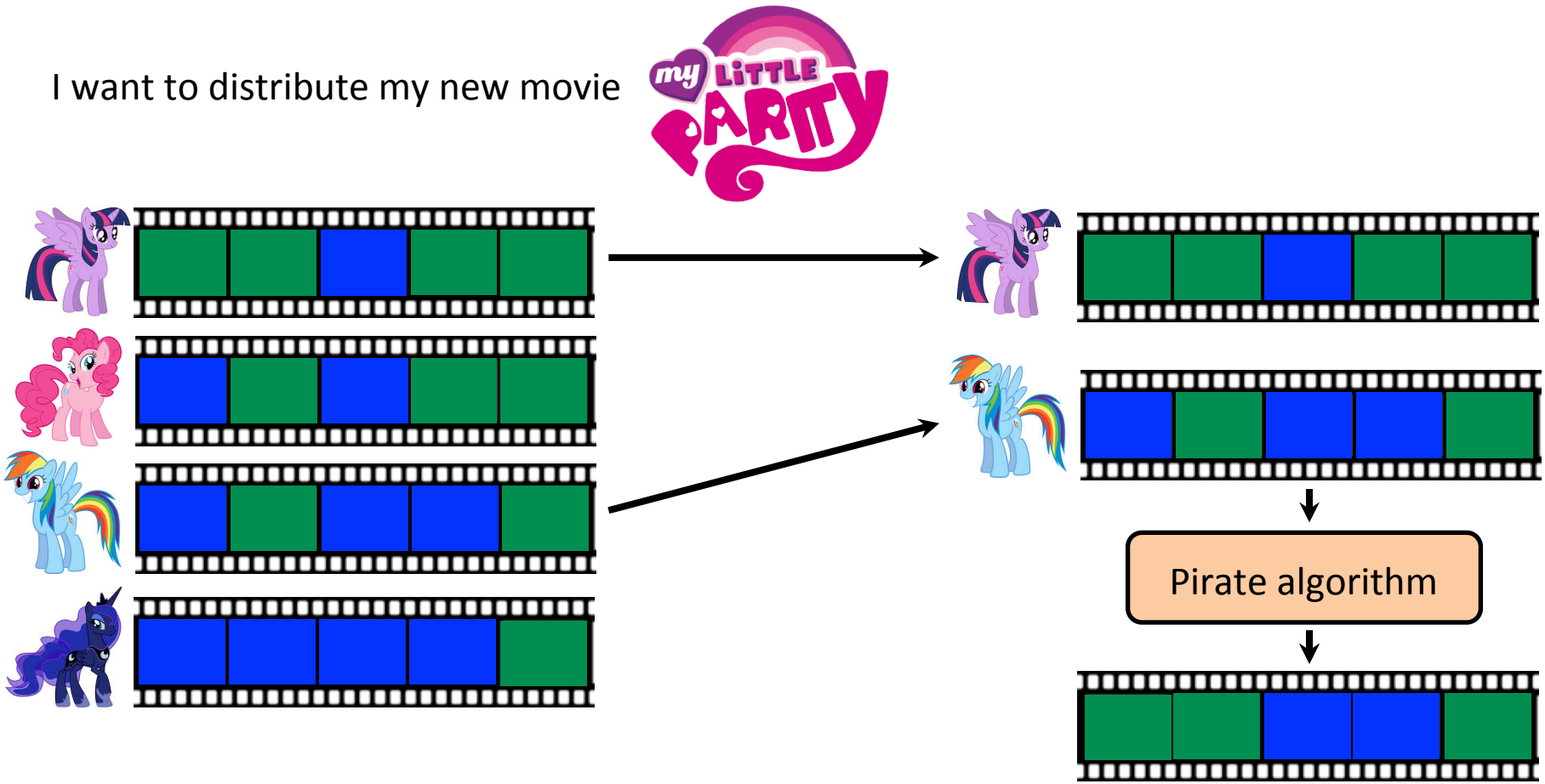


Trace
Algorithm

…but Equestria is full of pirates!

Fingerprinting Codes [Boneh-Shaw95]

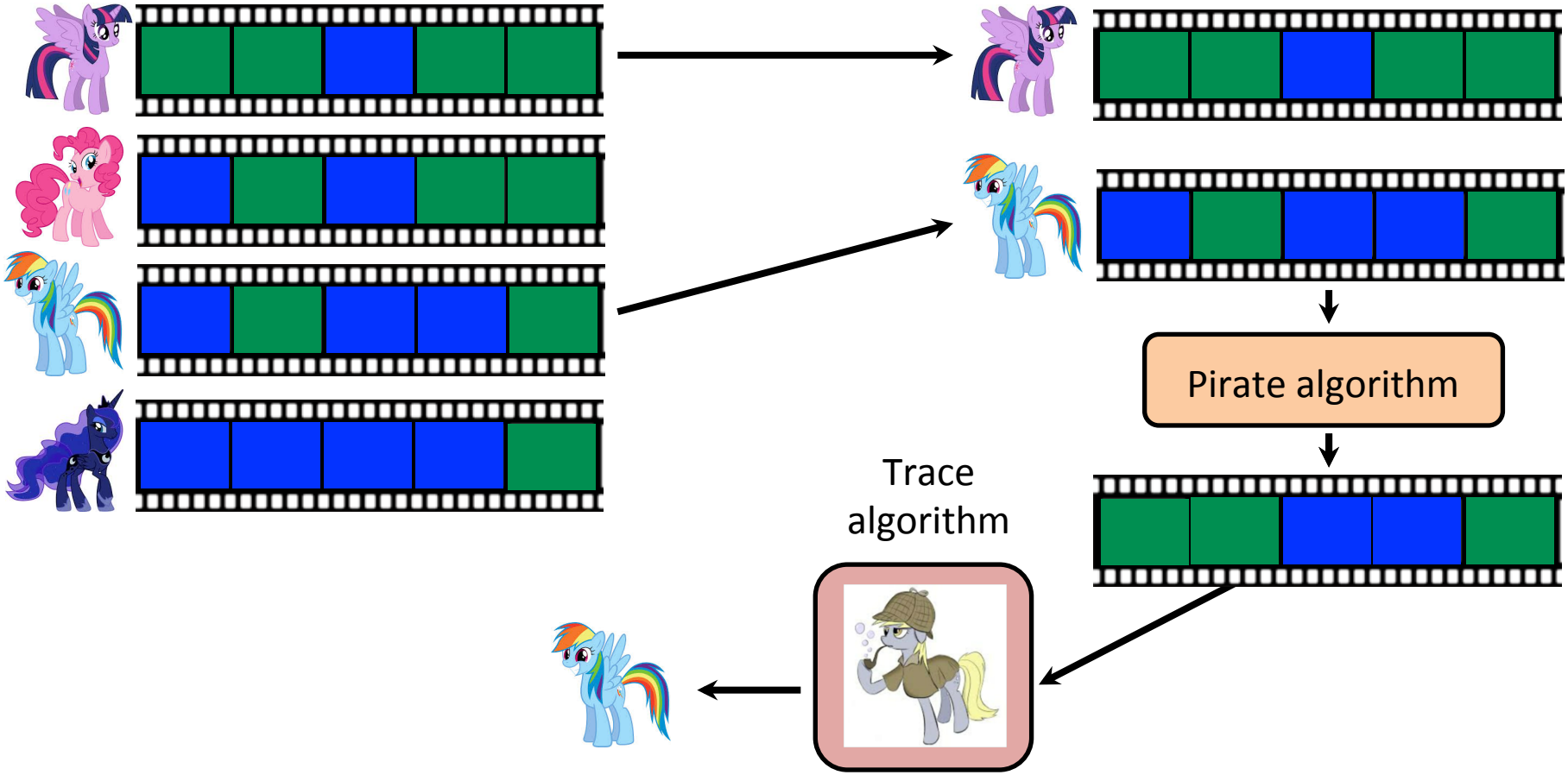
I want to distribute my new movie



…but Equestria is full of pirates!

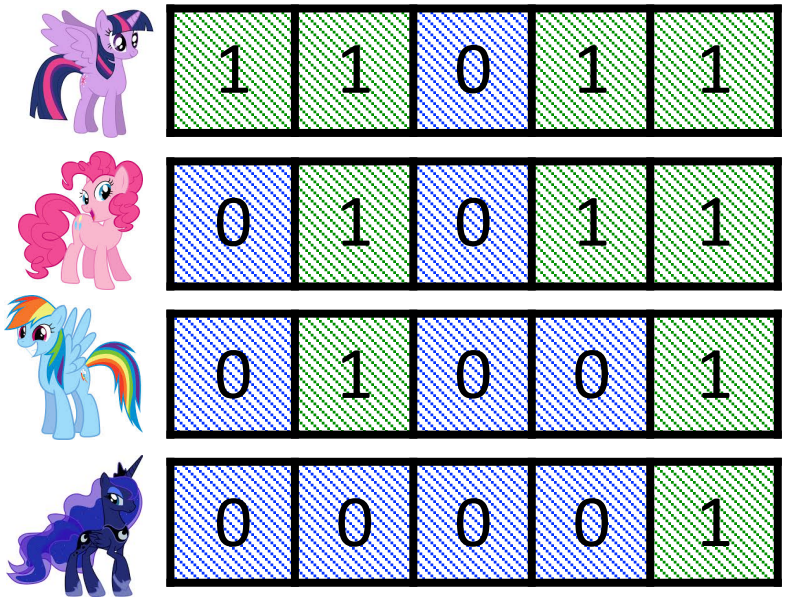
Who collude against me!

Fingerprinting Codes [Boneh-Shaw95]

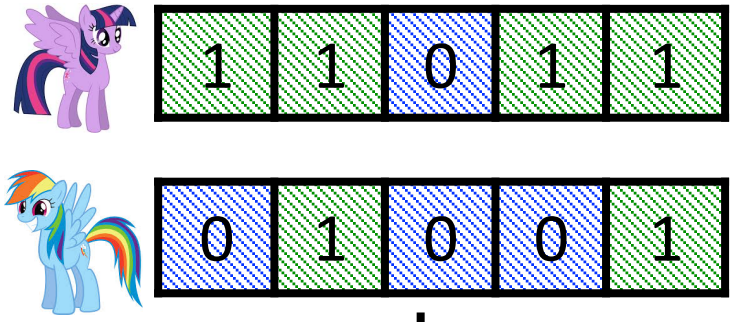


Fingerprinting Codes [Boneh-Shaw95]

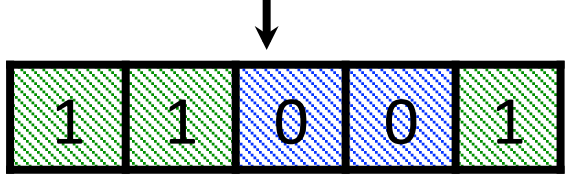
Gen(1^n) outputs $W \in (\{0,1\}^k)^n$



Pirate coalition $T \subseteq [n]$



Pirate algorithm



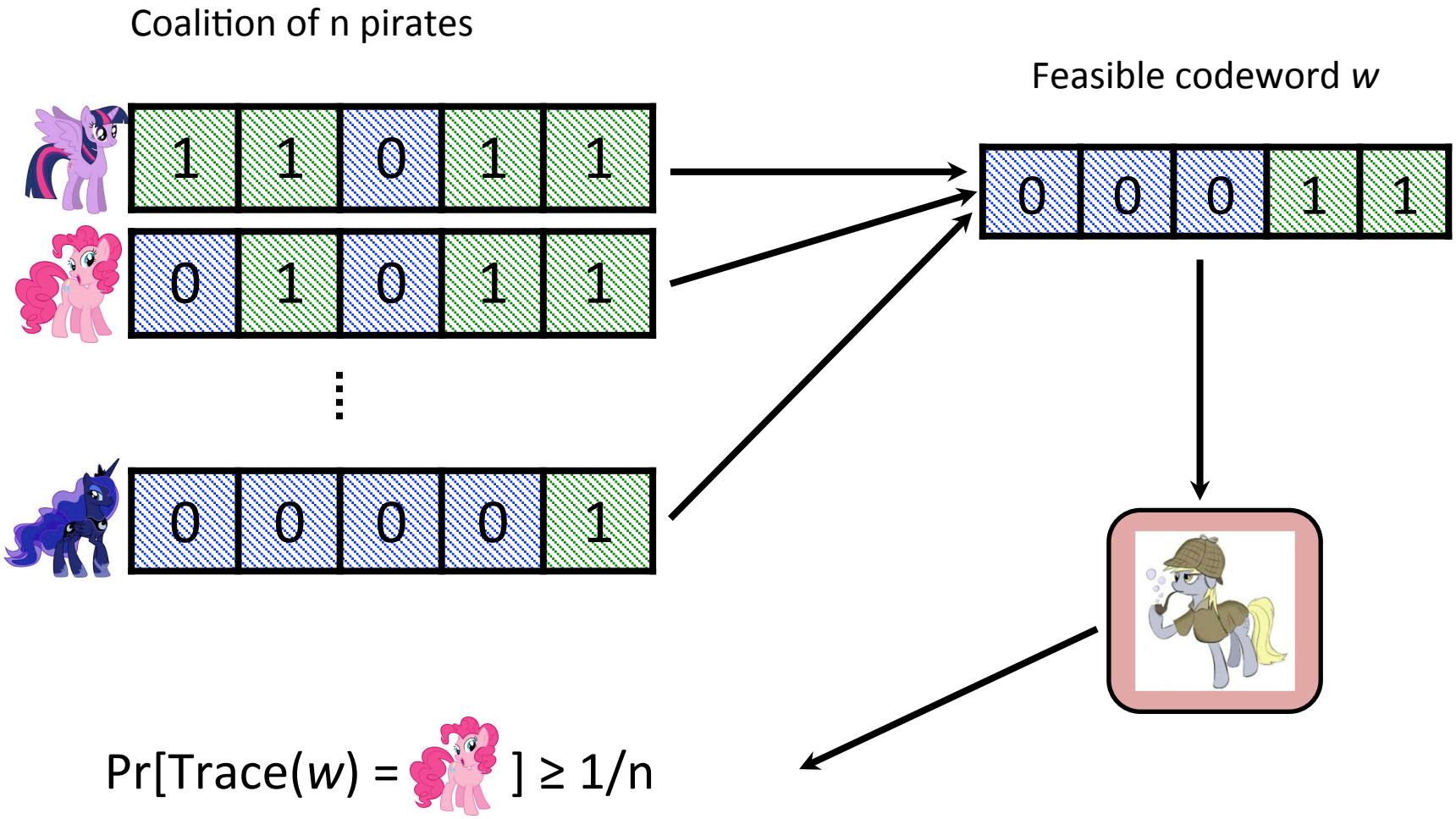
Feasible pirate codeword w

Trace algorithm

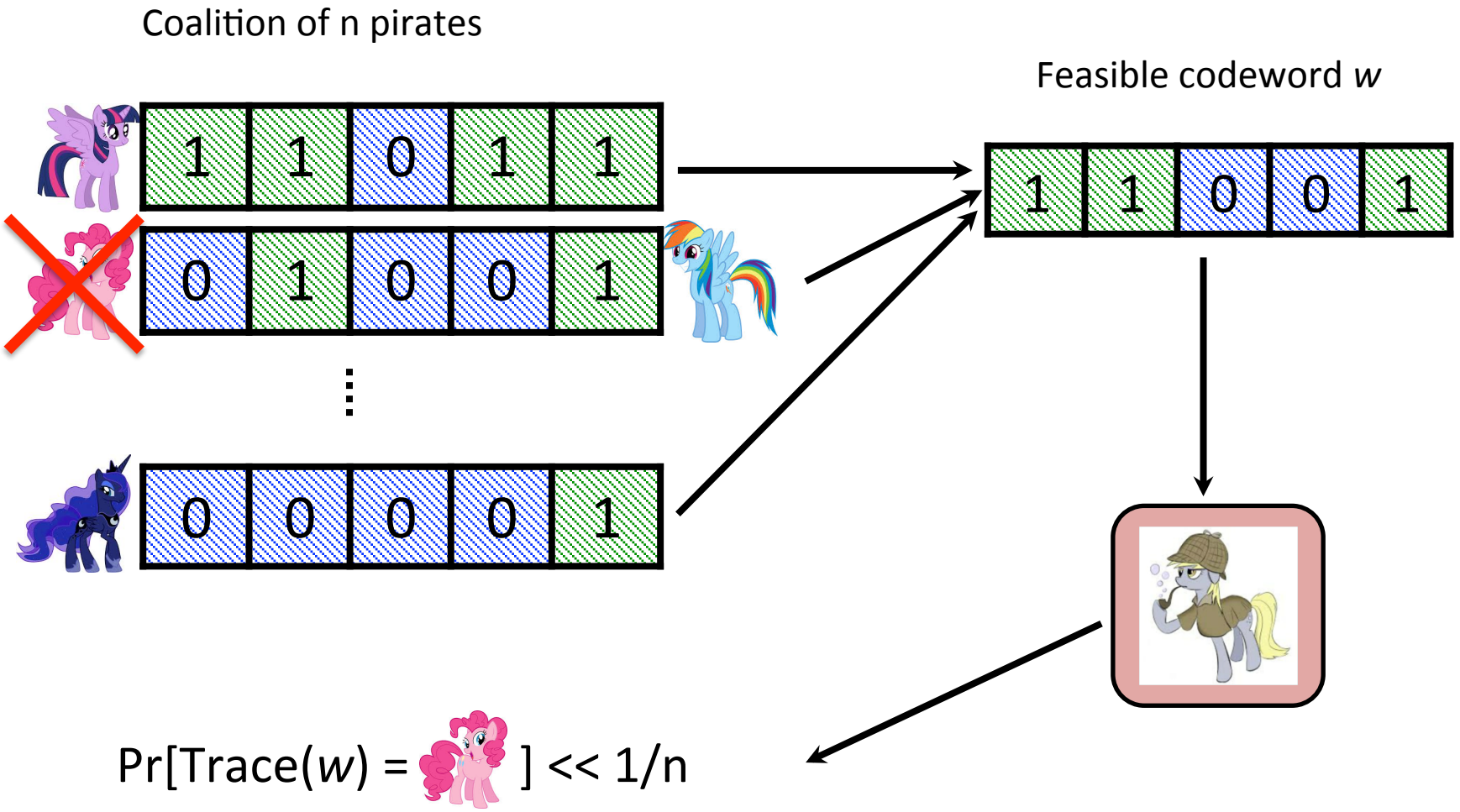
For all coalitions T and all pirate alg. for producing w,

$\Pr[\text{Trace}(w) \in T] \approx 1$

FP Codes vs. Diff. Privacy [B.-Ullman-Vadhan14]



FP Codes vs. Diff. Privacy [B.-Ullman-Vadhan14]



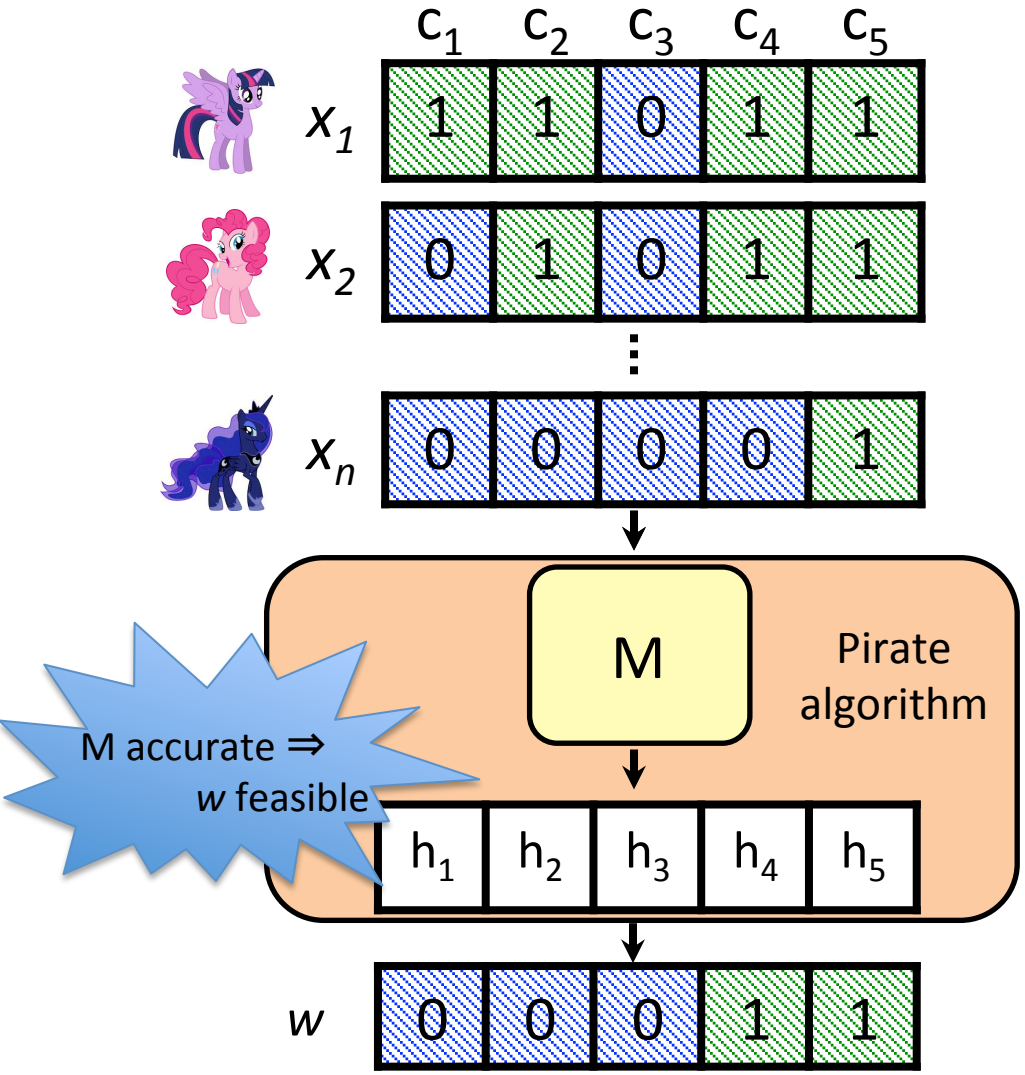
FP Codes vs. Diff. Privacy [B.-Ullman-Vadhan14]

Trace behaves very differently depending on whether  is in the coalition

Fingerprinting codes are the “opposite” of differential privacy!

Lower Bound for Thresholds

Labeled sample of n users = coalition of n users



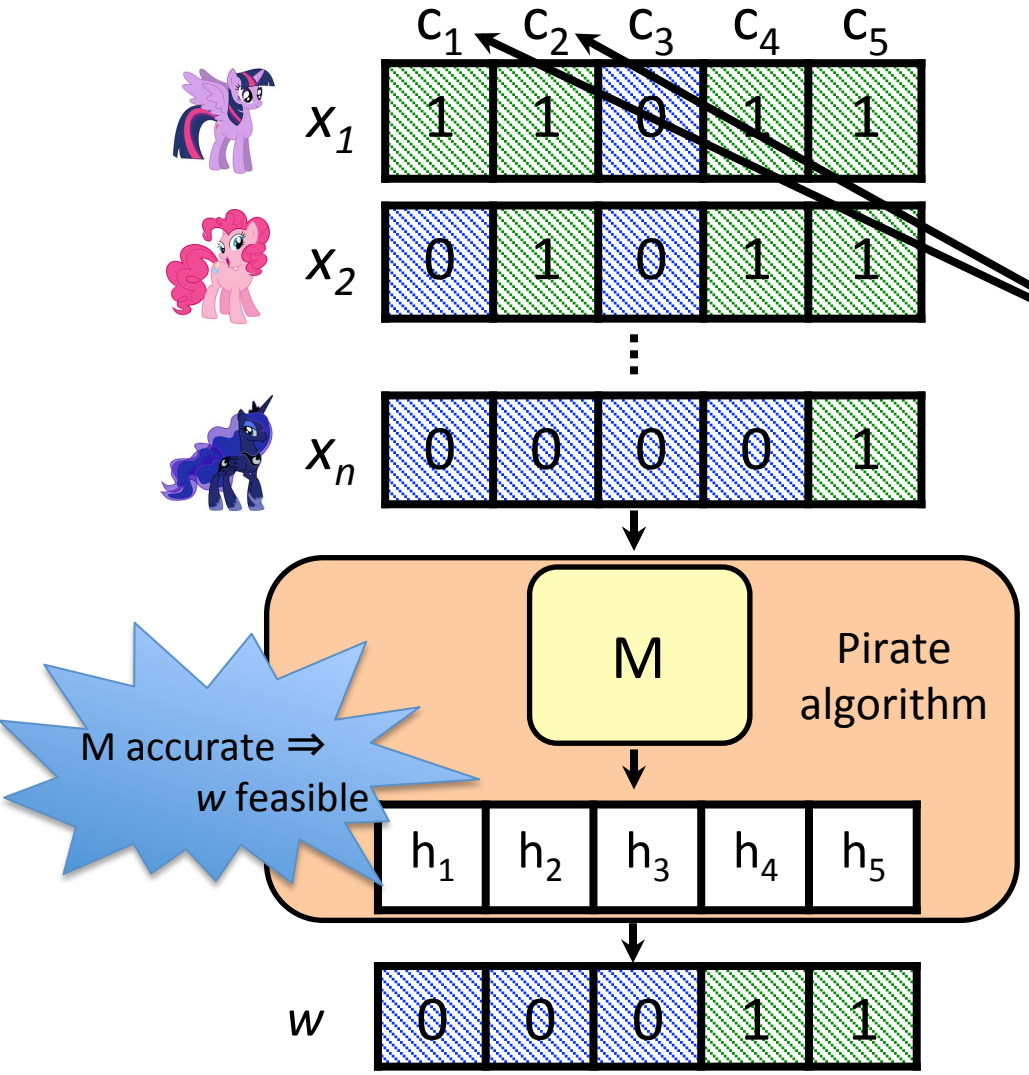
Suppose (for contradiction) we have

- A FP code of length k for (n+1) users
- A diff. private M that learns k threshold functions

Reduction: Use M to break security of the FP code

Lower Bound for Thresholds

Labeled sample of n users = coalition of n users



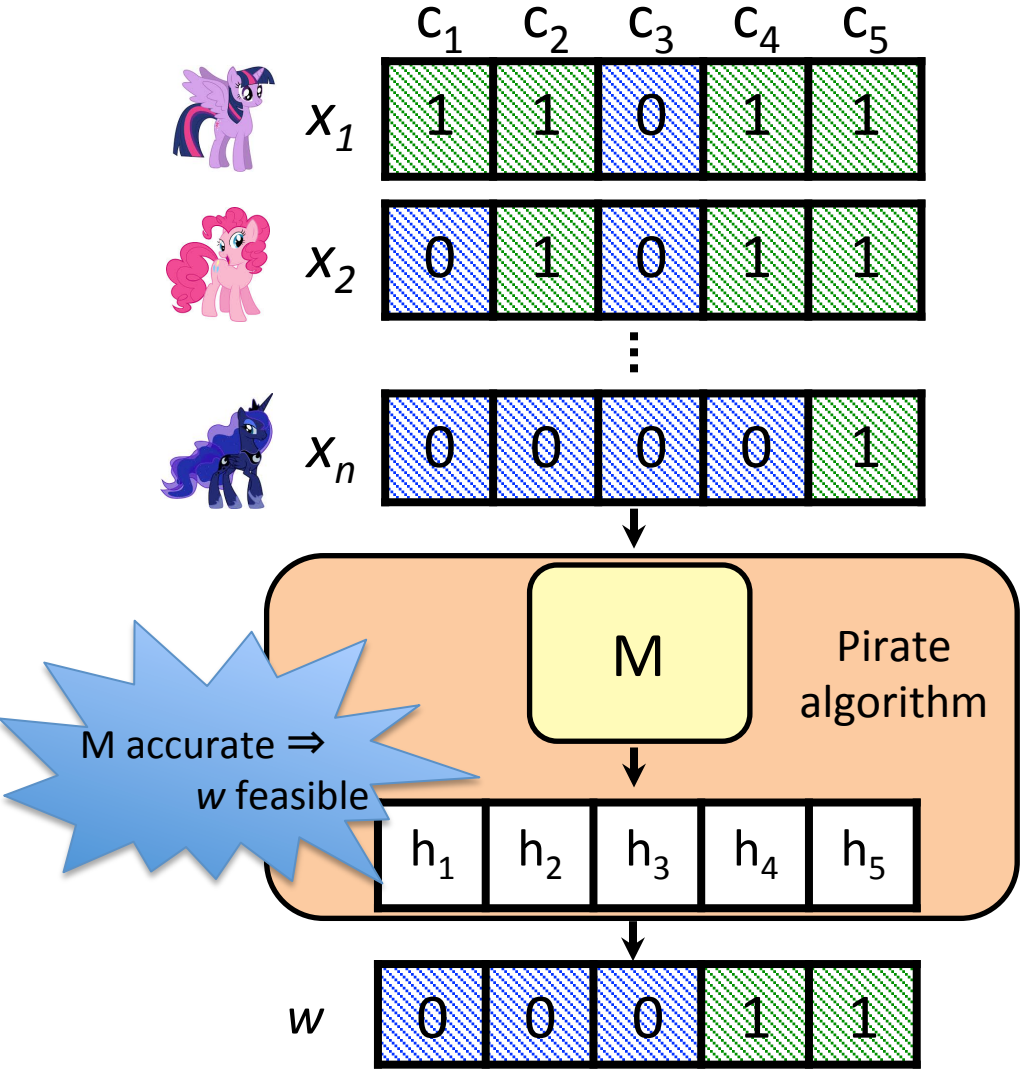
How do we ensure M is accurate?

Each column of the codebook needs to be consistent with a threshold concept

Magic observation:
The FP code of [BS95] has this structure

Lower Bound for Thresholds

Labeled sample of n users = coalition of n users



Suppose (for contradiction) we have

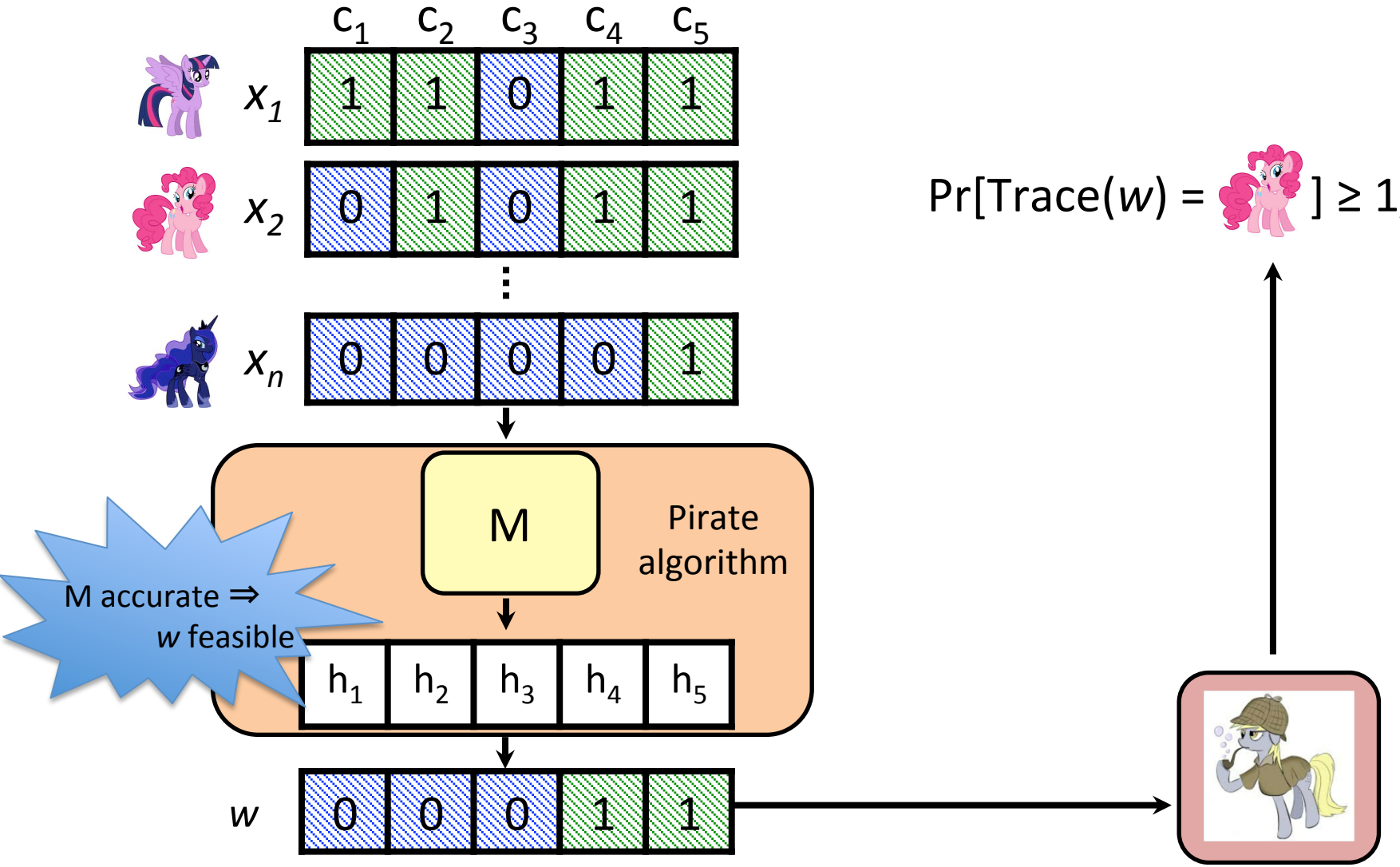
- A nice FP code of length k for (n+1) users
- A diff. private M that learns k threshold functions

Reduction: Use M to break security of the FP code

$$w_j = round\left(\frac{1}{n}\sum_{i=1}^{n} h_j(x_i)\right)$$

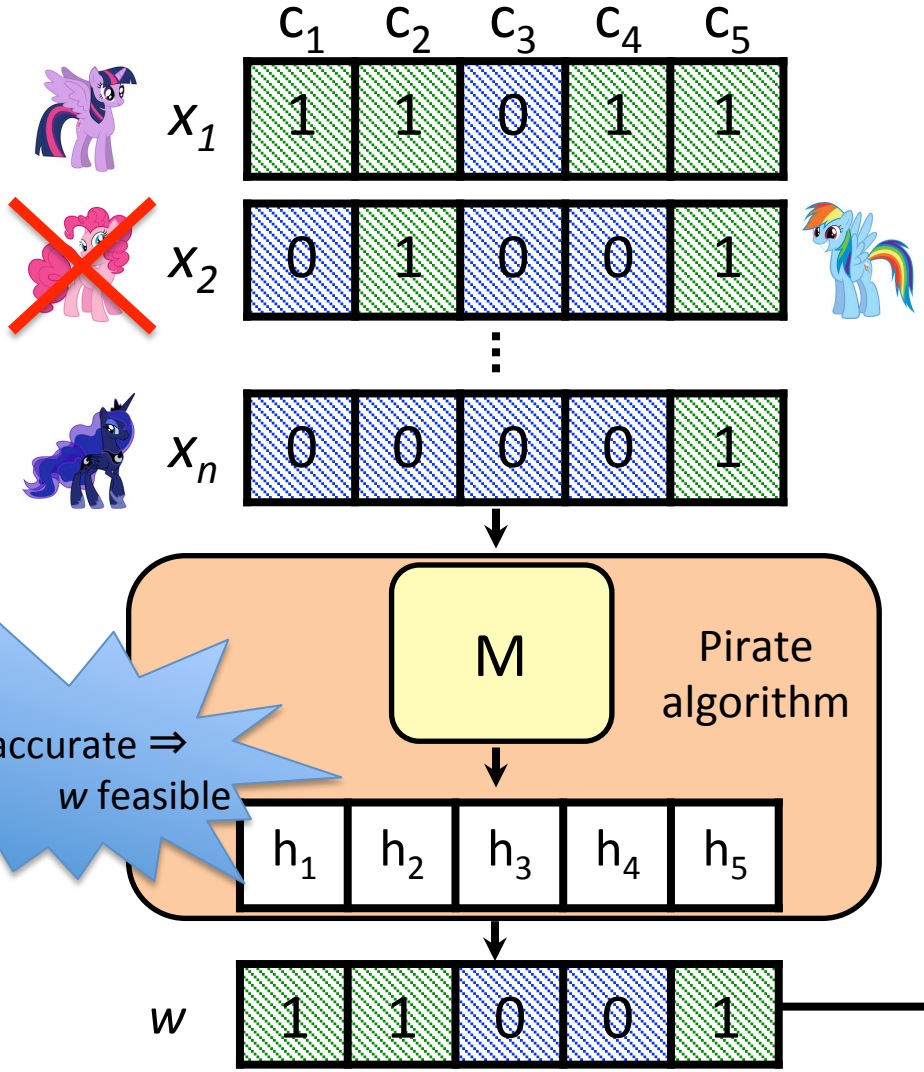
Lower Bound for Thresholds

Labeled sample of n users = coalition of n users



Lower Bound for Thresholds

Labeled sample of n users = coalition of n users



Contradicts security of FP code!

$$\Pr[\text{Trace}(w) = \text{Pinkie Pie}] \geq \frac{(1/n) - \delta}{1+\varepsilon} \geq \frac{1}{3n}$$

M private ⇒ Trace fails

Lower Bound for Thresholds

- ∃ "nice" FP code for n users with length k
  $\Rightarrow$ learning k thresholds requires n samples

- [BS95] ∃ "nice" FP code for $\Omega(k^{1/3})$ users of length k
  $\therefore$ learning k thresholds requires $n \geq \Omega(k^{1/3})$

Conclusions

- Introduce study of private multi-learning
- Paint a complex picture of how sample complexity depends on k

- Open questions
  - Is dependence on poly(k)VC(C) necessary?
  - Other examples of "direct-sum" tasks?

Generic Multi-Learner

- Apply technique from [Beimel-Nissim-Stemmer15] for reducing labeled sample complexity

- Idea: 1. Identify set H of $2^{VC(C)}$ "important" concepts via sanitization
  2. Run [KLNRS08] generic learner k times using H as hypothesis class

- Total sample complexity =
  fixed cost of sanitization + $k^{1/2}$ VC(C)