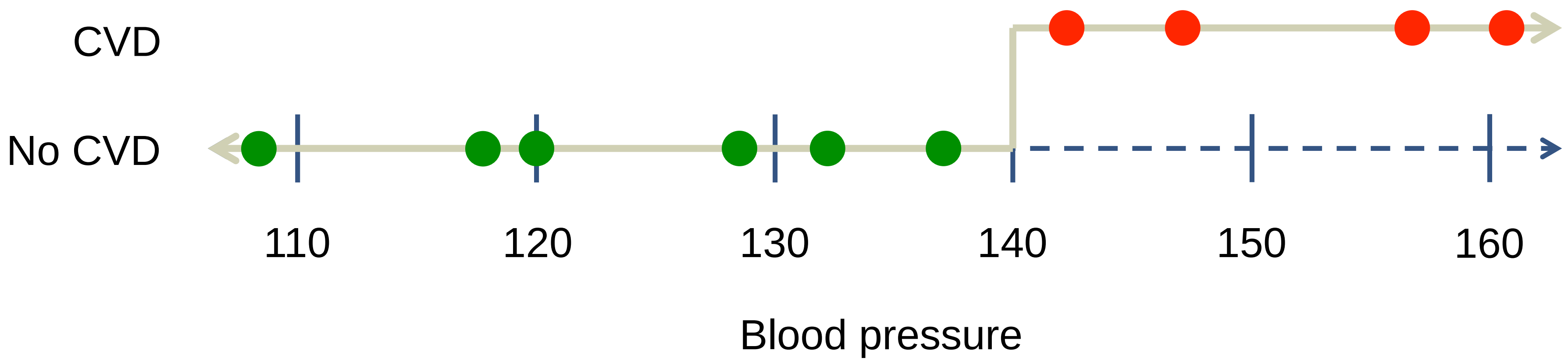
Order-Revealing Encryption and the Hardness of Private Learning

Mark Bun – Harvard University
Mark Zhandry – Stanford University

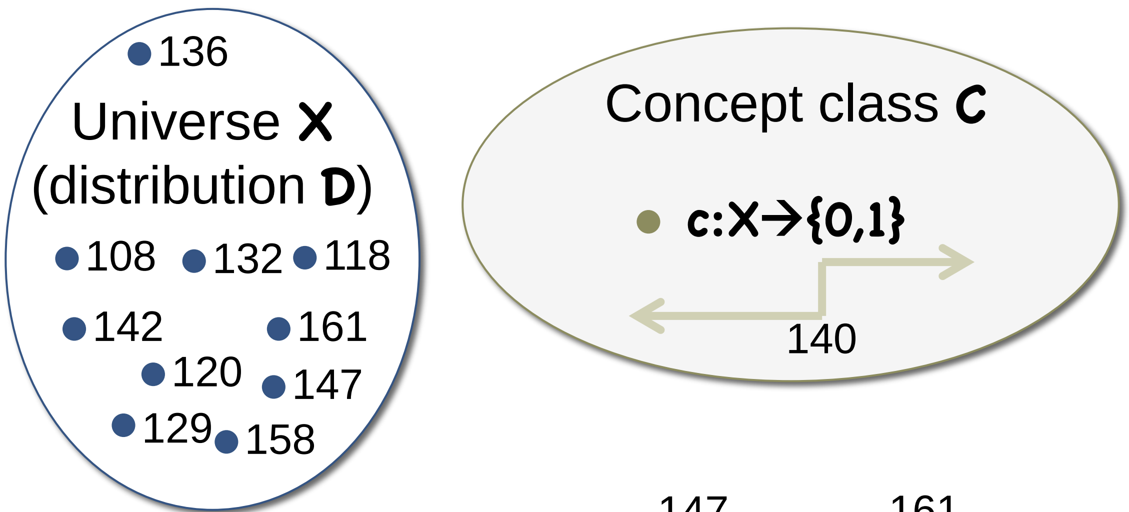
Example: Learning from Patient Data

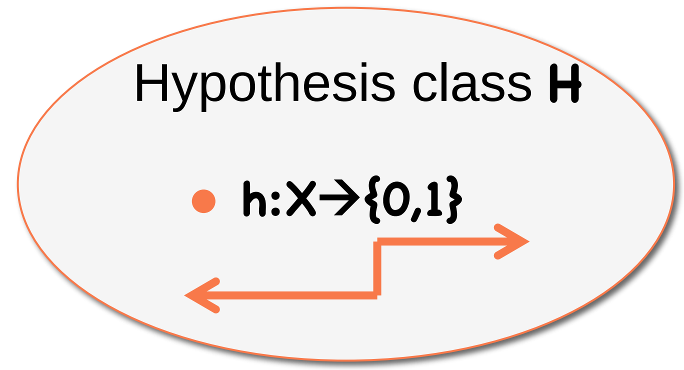


Goals:

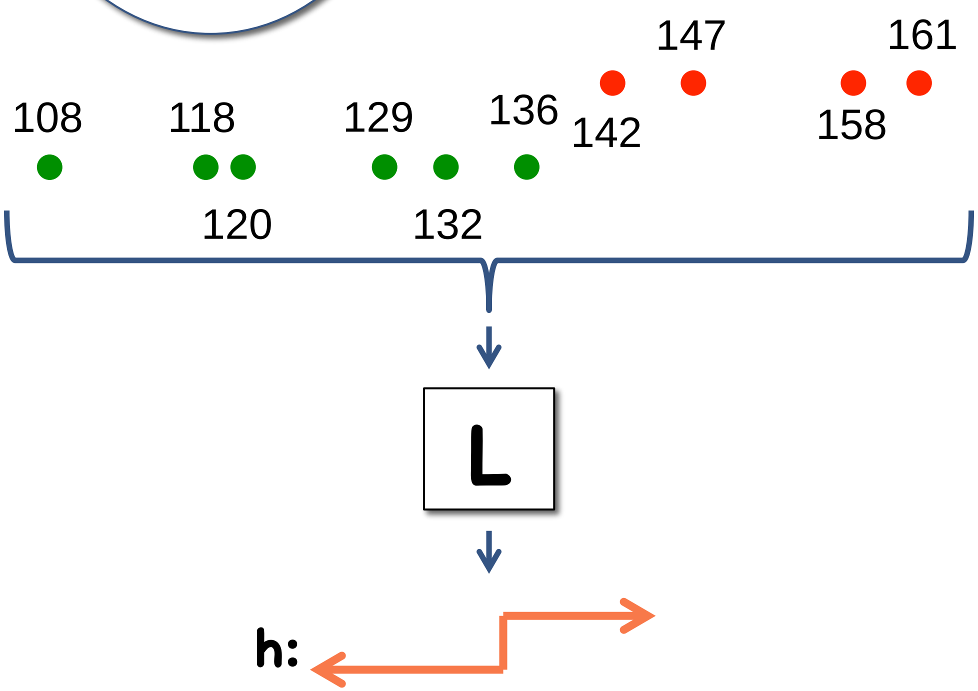
Learn threshold　　Maintain privacy

(Distribution free) PAC Learning [Val'84]





(**H=C**: proper learning)



h is α-approx. correct if:

Pr[h(x)=c(x)] ≥ 1- α

where **x←D**

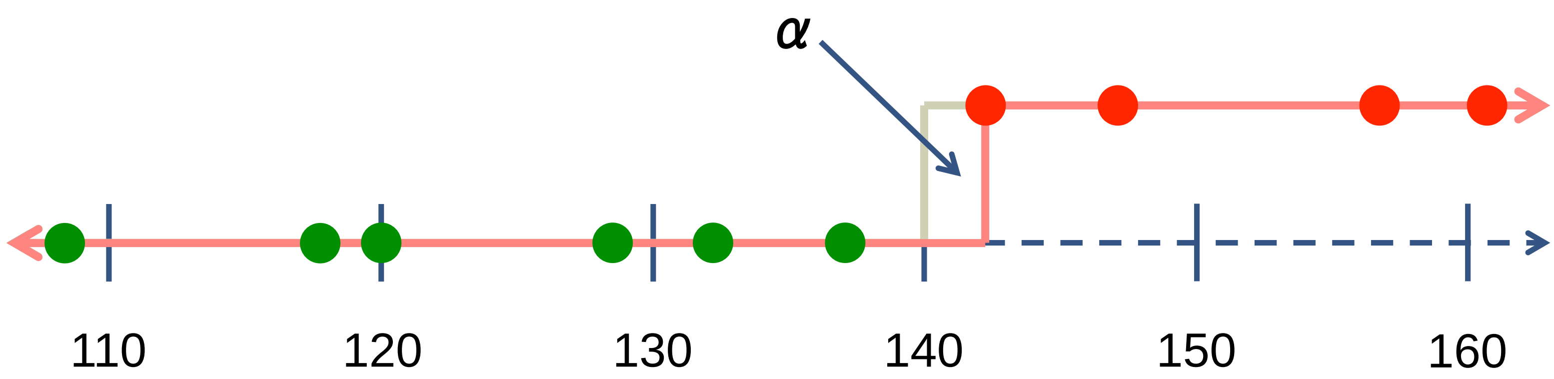
L is **(** α **,** β **)**-accurate if:

Pr [h is α **-AC] ≥ 1-** β

<u>for all **D**</u>

How do we learn threshold?

Answer: threshold at smallest positive sample



Fact: O($\log(1/\beta)/\alpha$) samples ⇒ **(α, β)**-accurate

Learnability in General

Fact: Any ***C*** can be properly learned using **$O(\log |C|)$** samples

"Occam's Razor": Pick ***c*** consistent with all samples

- Problem: running time **$O(|C|)$**, exponential in description size
- Learner not efficient

Only few efficient learning algorithms

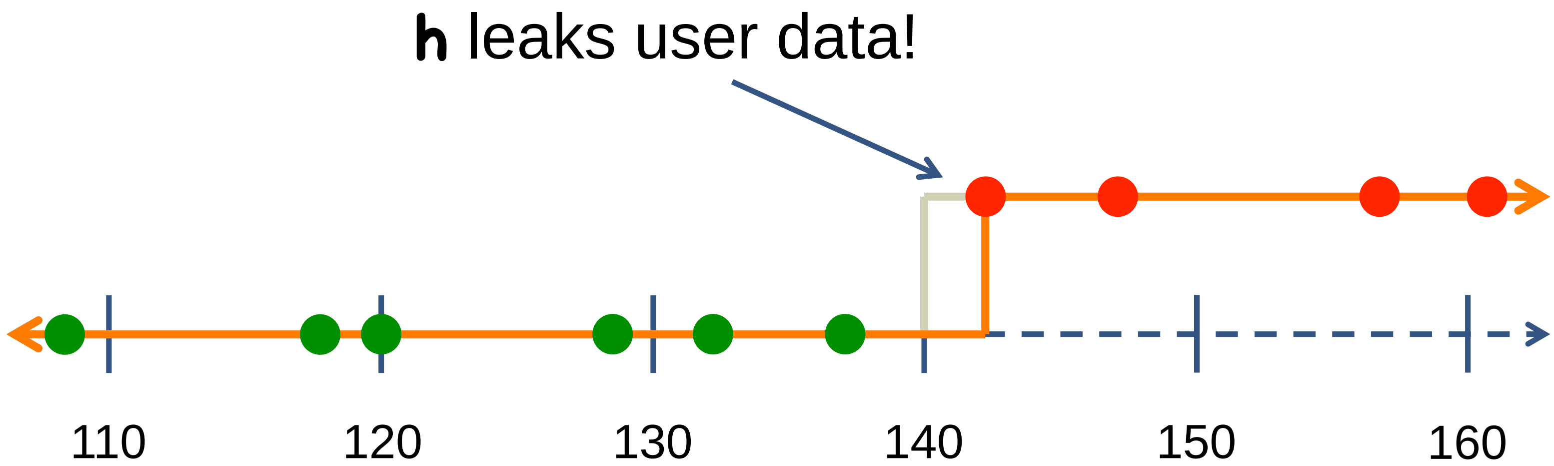
- Statistical query learning [Kea'98], Gaussian elimination

There are problems that cannot be learned efficiently*

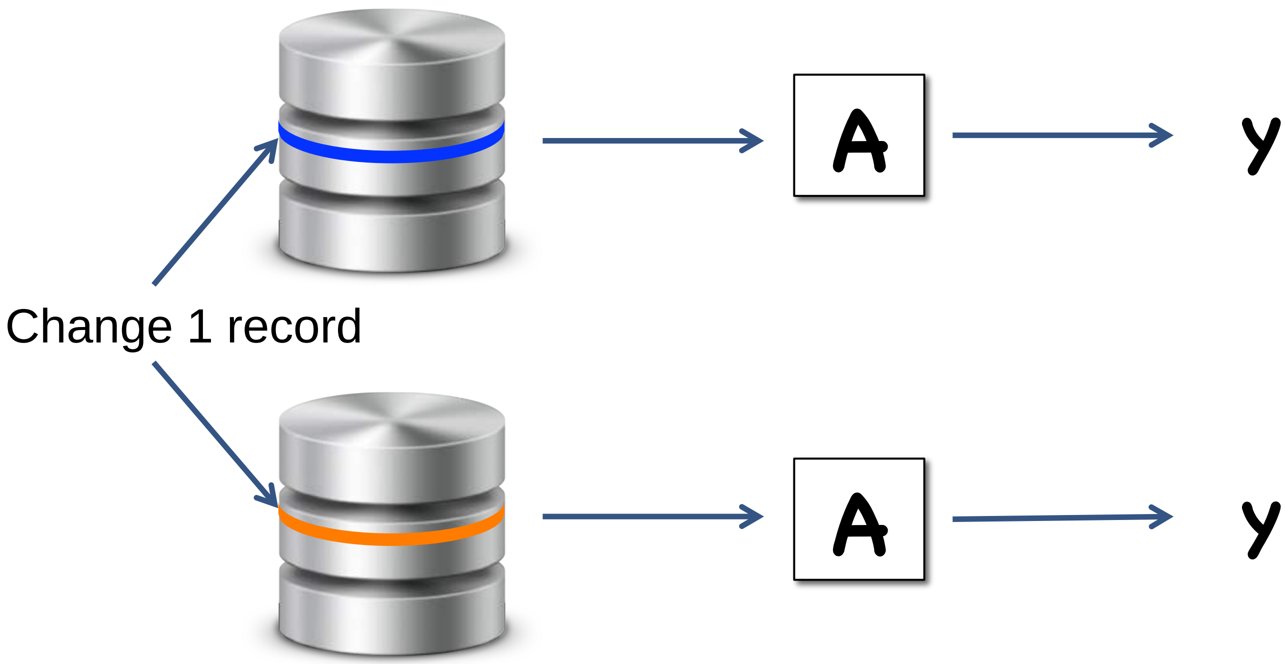
- e.g. PRFs

*under reasonable assumptions

Privacy Problem



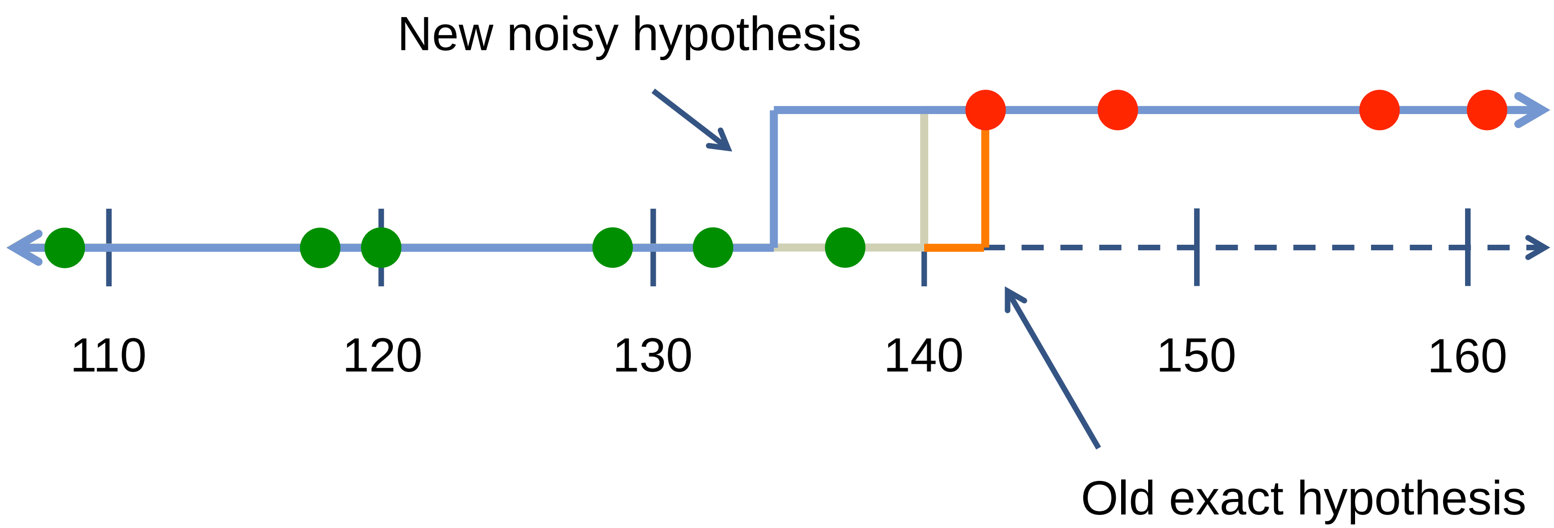
Differential Privacy [DMNS'06]



Differential Privacy ⇒ output distributions are "close"

A Differentially Private Threshold Learner

Solution: add noise!



Learning and Differential Privacy

Thm ([KLNRS'11]): Any C can be privately learned using $O(\log |C|)$ samples

"Private Occam's Razor":

- Sample random c weighted according to accuracy
- Again, learner not efficient

Statistical query, Gaussian elimination can be privately simulated

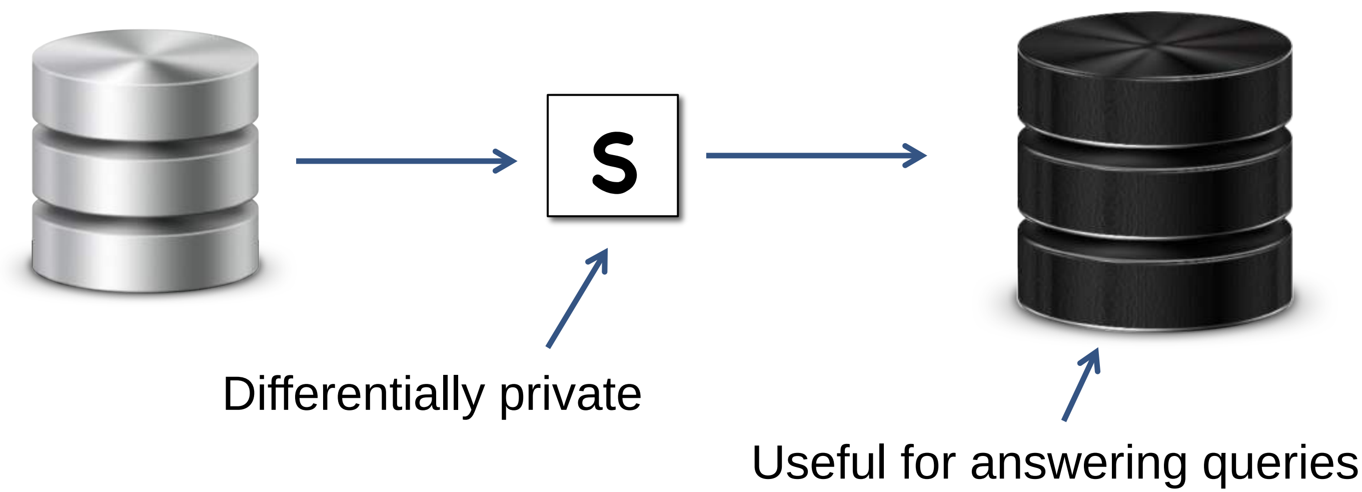
- [BDMN'05], [KLNRS'11]

Question ([KLNRS'11]): Are all efficiently learnable concepts efficiently privately learnable?

Answer: No

Crypto and Differential Privacy

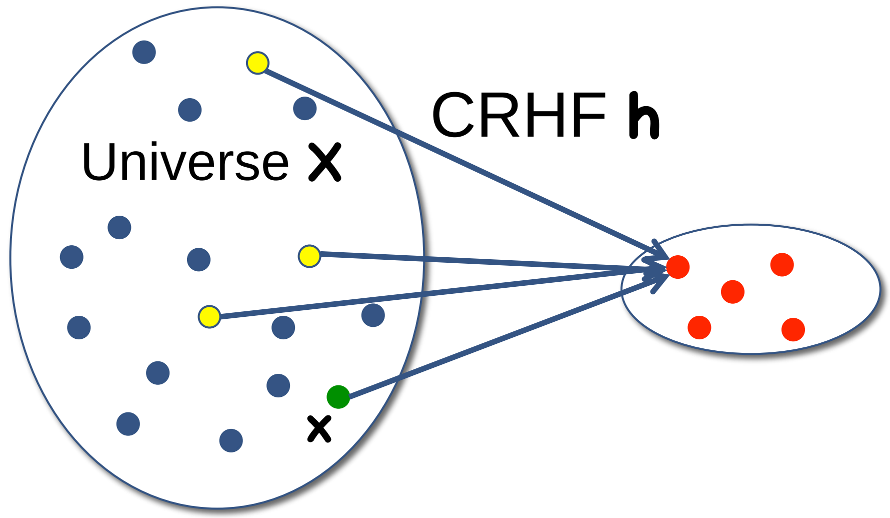
Example: private data release



Thm ([DNRRV'09], informal): Traitor tracing ⇒ impossibility for private data release

[GGHRSW'13, BZ'14]: Traitor tracing form iO

Partial Result: Proper Learning [Nis'14]



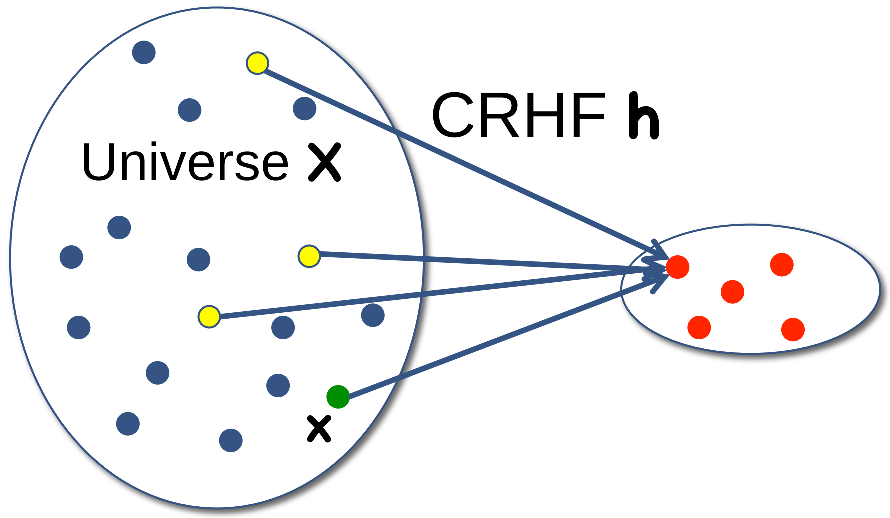
$$C = H = \{f_x(y):\ h(x)=h(y)?\}$$

Any positive sample **x** is a representation of f_x
⇒ C is efficiently properly PAC learnable

Given some positive samples, infeasible to find new rep.
⇒ Cannot privately PAC learn a representation **x**

Can be based on any OWF

Partial Result: Proper Learning [Nis'14]



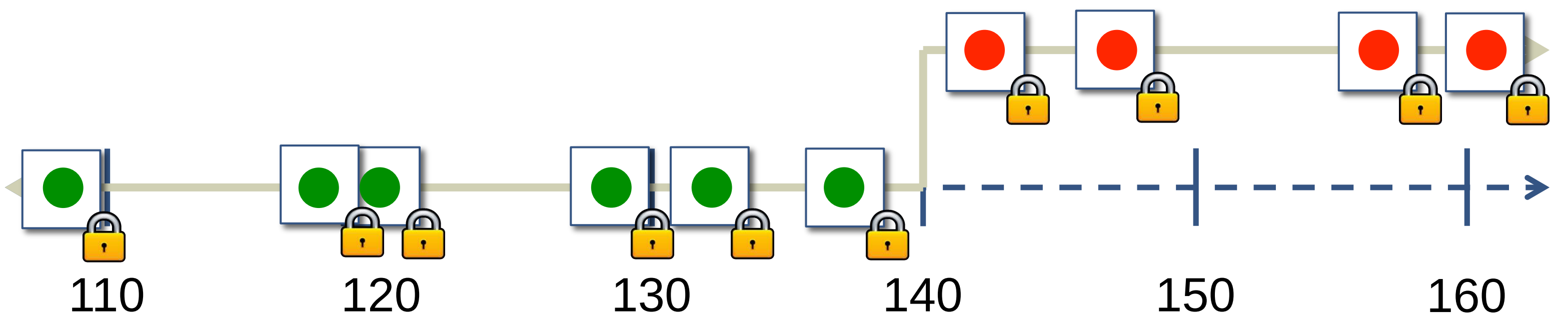
$$C = H = \{f_x(y):\ h(x)=h(y)?\}$$

Limitation: **x** is not the only representation of f_x as a function

- $g_z(y)$: $h(y)=z$? where $z=h(x)$
- Can privately properly learn representation **z**
- Counterexample only applies to "representation learning"

Question: How to extend this to general (non-proper) learning?

Idea: Encrypted Threshold



Universe = ciphertexts

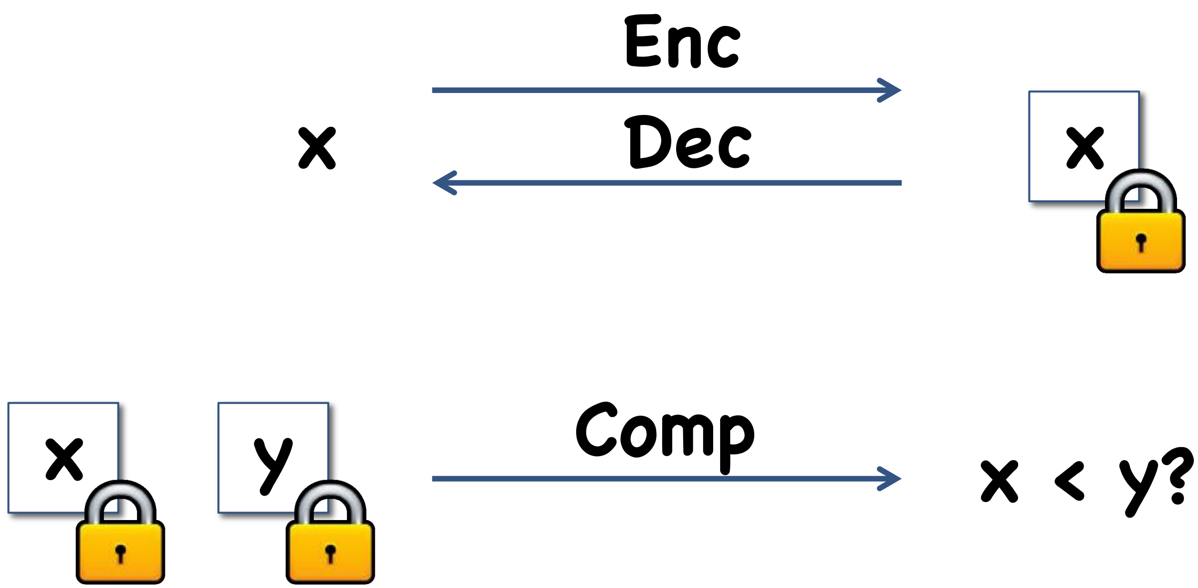
$f_t(c)$: Dec(c) ≥ t?

Question: How to learn?

Observation: Threshold learner only needs to know order of data

Order Revealing Encryption [BCLO'09, PR'12]

Encryption where order is revealed, but nothing else



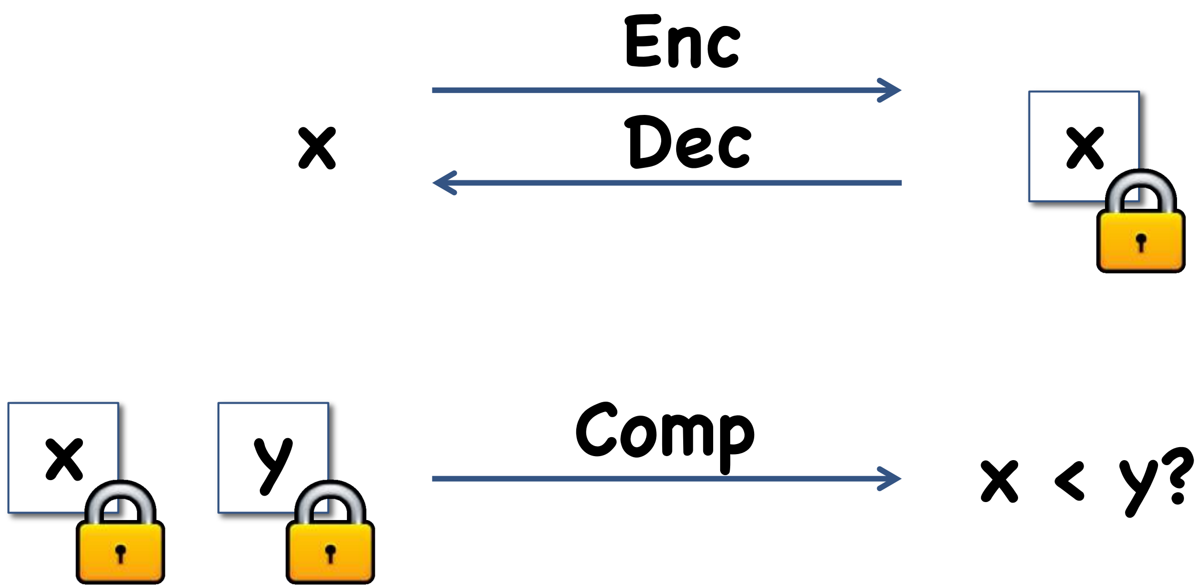
Enc,Dec secret
Comp public

~~Weak correctness: for any **x,y**,~~

~~$$\Pr[\text{Comp}(\text{Enc}(x), \text{Enc}(y)) = (\, x < y?\,)\,] = 1$$~~

Order Revealing Encryption [BCLO'09, PR'12]

Encryption where order is revealed, but nothing else



Enc,Dec secret
Comp public

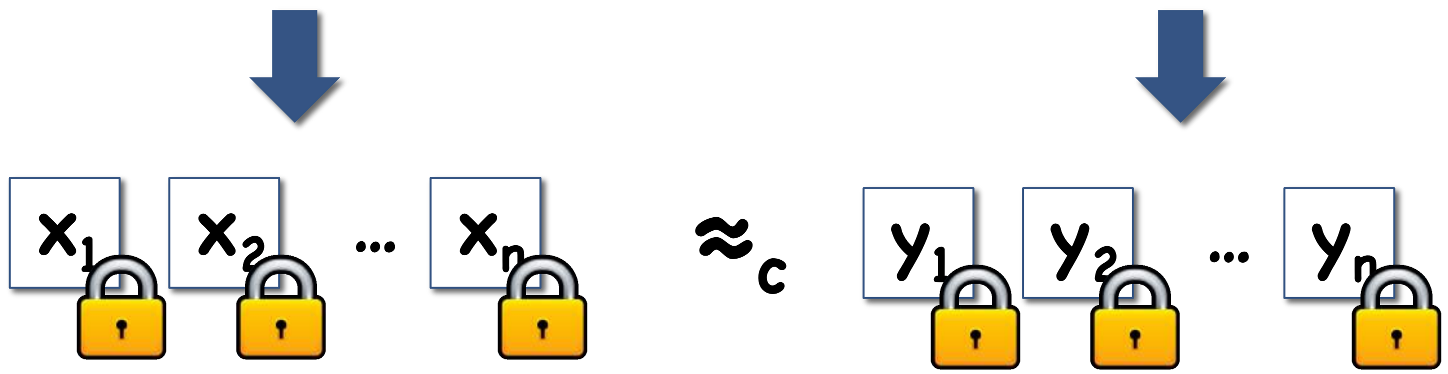
<u>Strong</u> correctness: for any c_0, c_1,

$$\Pr[\mathrm{Comp}(c_0,c_1) = (\ \mathrm{Dec}(c_0) < \mathrm{Dec}(c_1)?\ )\ ] = 1$$

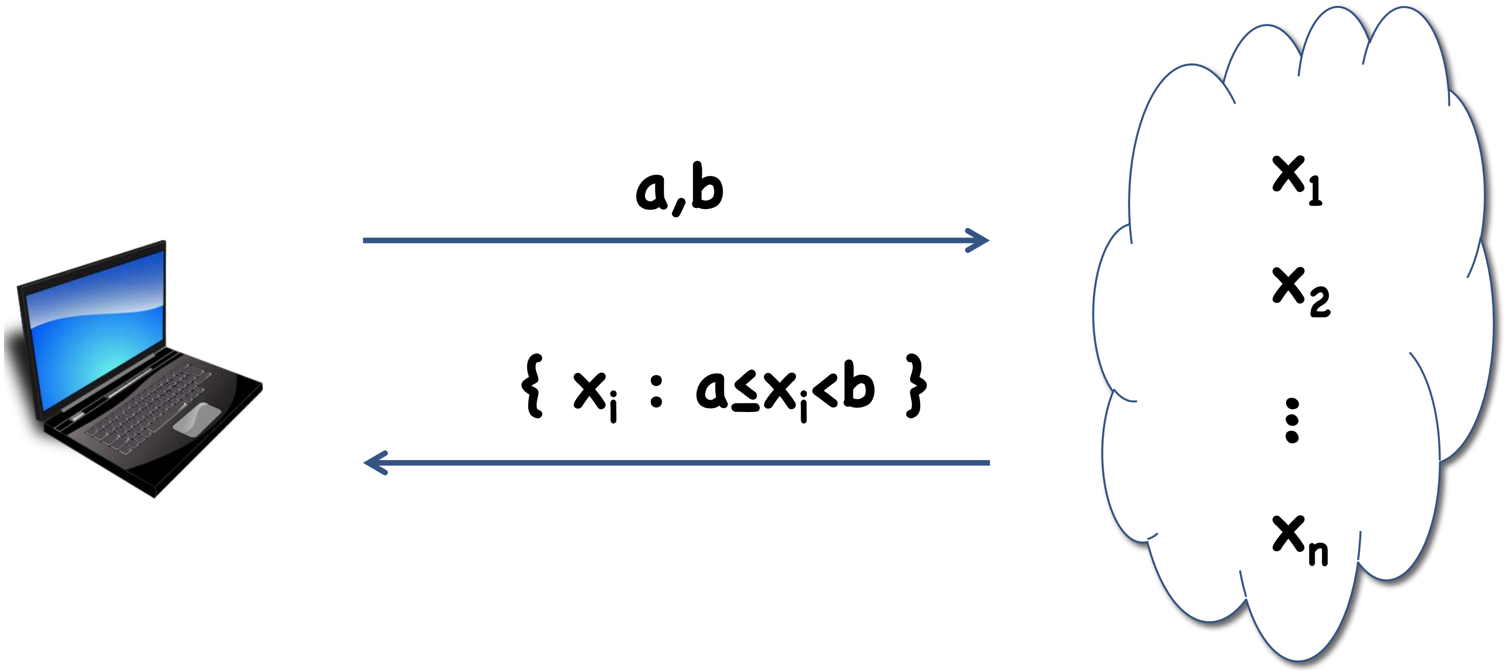
ORE Security

"Best possible" security: only order revealed

$x_1 < x_2 < \ldots < x_n$ $\qquad$ $y_1 < y_2 < \ldots < y_n$

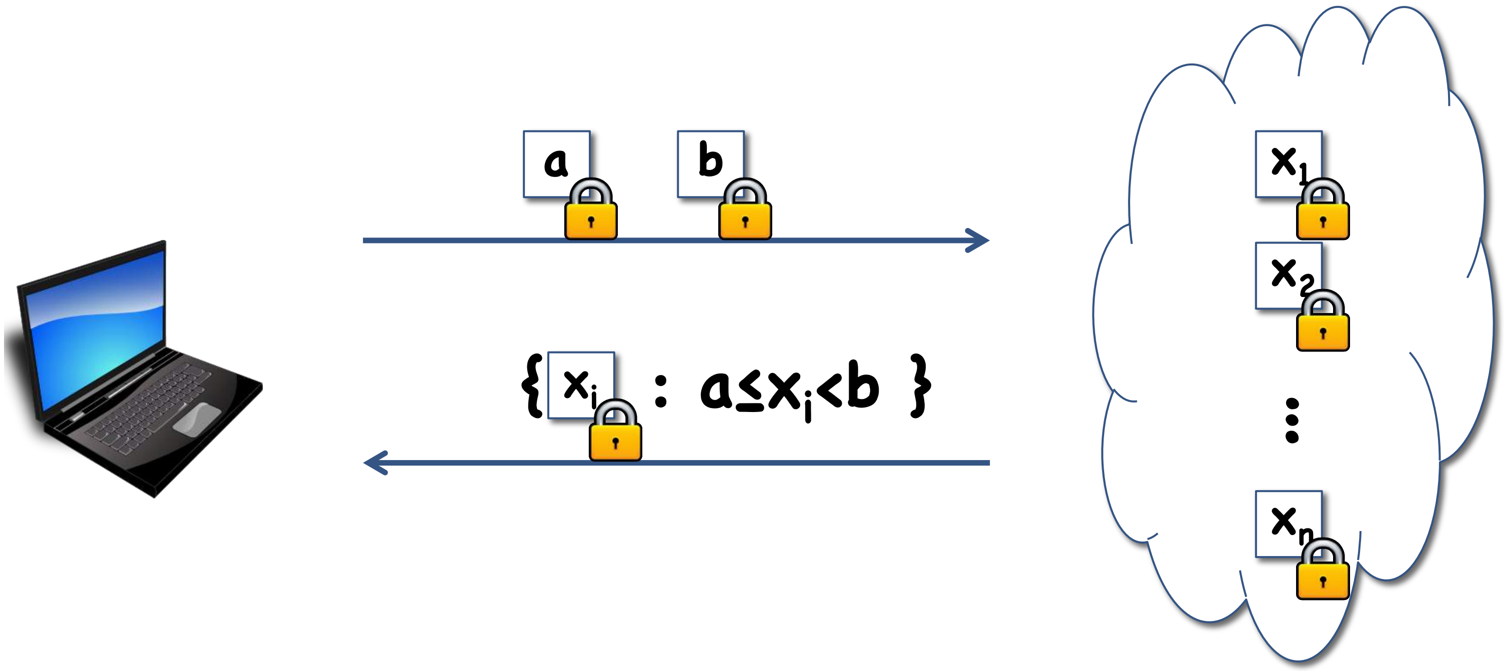


ORE for Encrypted Range Queries



Goal: Hide database and query from cloud

ORE for Encrypted Range Queries



ORE vs OPE

OPE = Order *preserving* encryption [BCLO'09]

- Ciphertext space is totally ordered
- Decryption is monotonic (so **$Comp(c_0,c_1) = (c_0 < c_1 ?)$**)
- OPE cannot obtain "best possible" security
- Much weaker notion: indist. from rand. monotonic function
- Can build from one-way functions

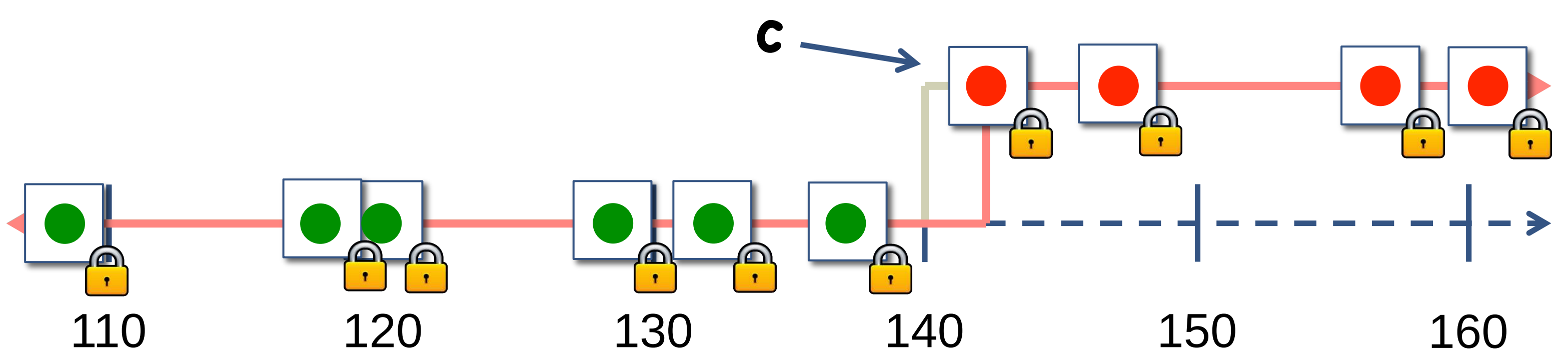
ORE:

- Much weaker correctness requirement
- Much stronger security requirement
- Will discuss constructions shortly

Learning Encrypted Threshold

Still threshold at smallest positive (encrypted) sample

- Hypothesis uses ctxt comp. instead of ptxt comp.



$$h_c(c') = \mathrm{Comp}(c, c')$$

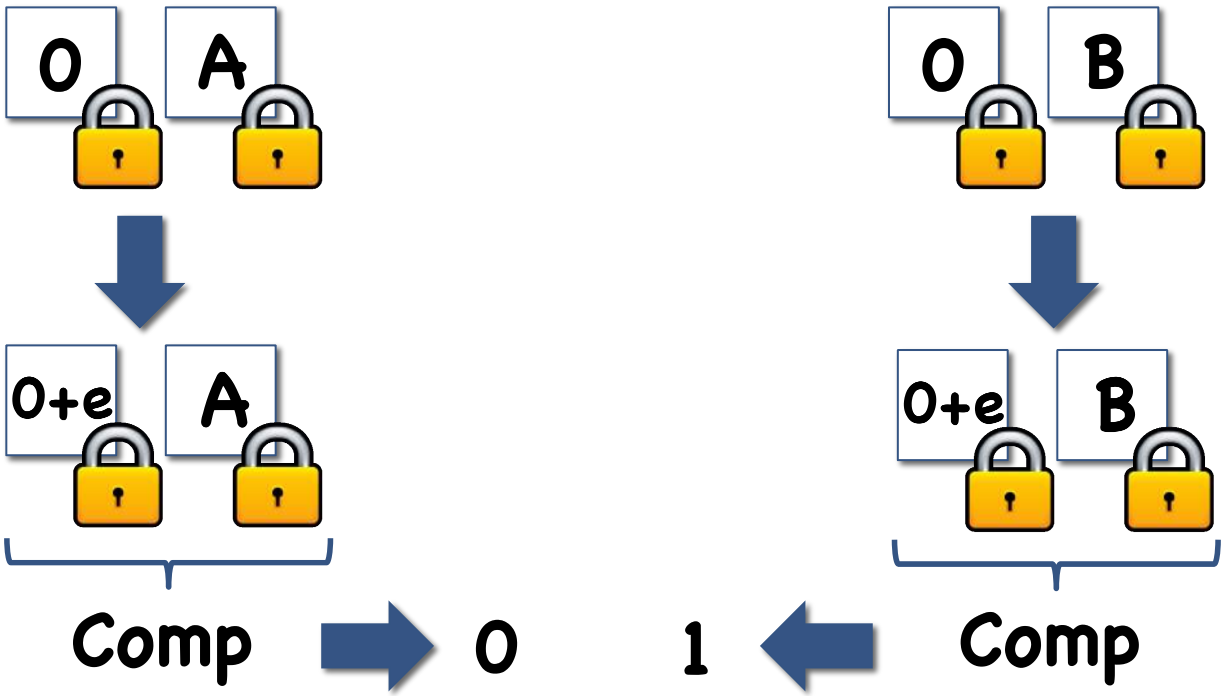
Thm: Encrypted threshold is efficiently PAC learnable

What about private learning?

Private Learnability of Encrypted Threshold

Intuition: ORE is non-malleable, so can't add noise

- Proof by contradiction: suppose possible to add noise $e \in [A,B)$

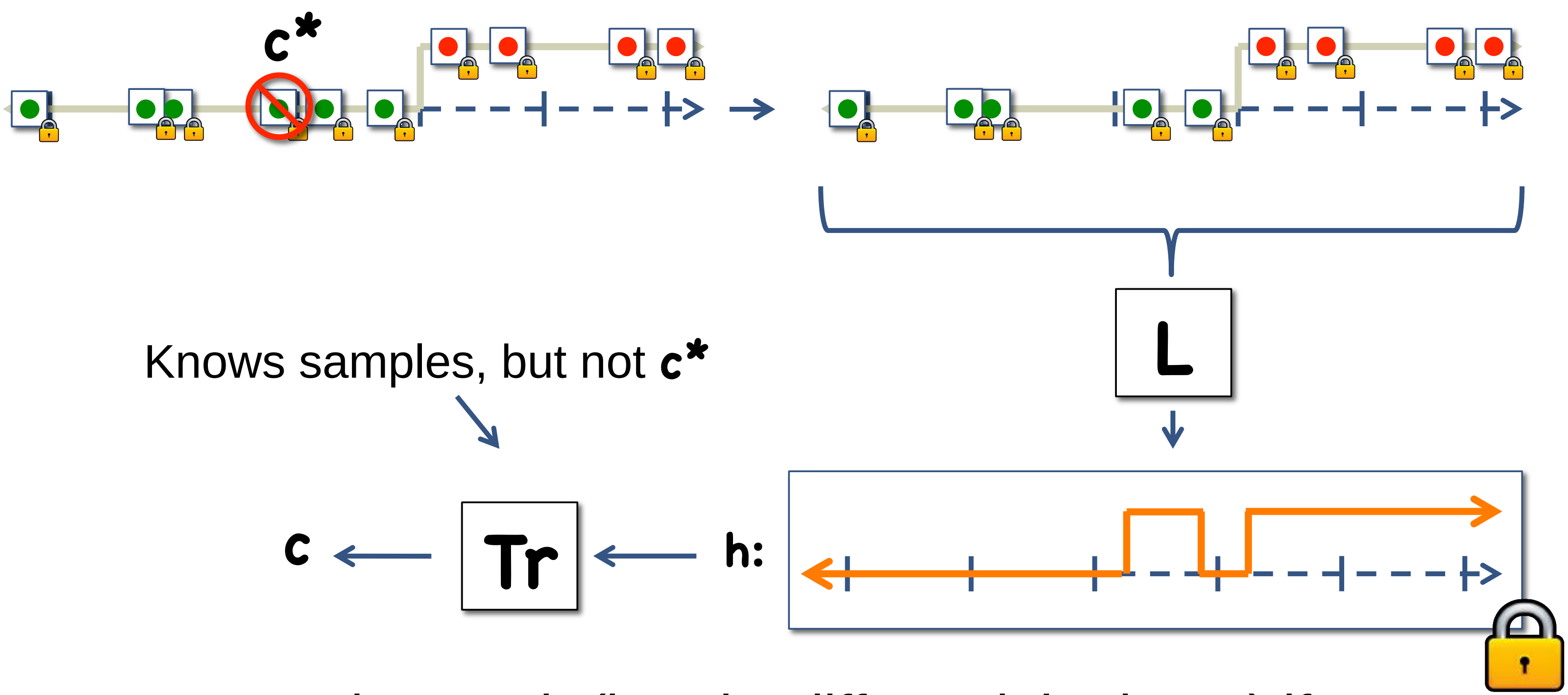


Question: how to formally prove private learning is impossible?

- Difficulty: no restrictions on form of hypothesis

Re-identification for Encrypted Threshold

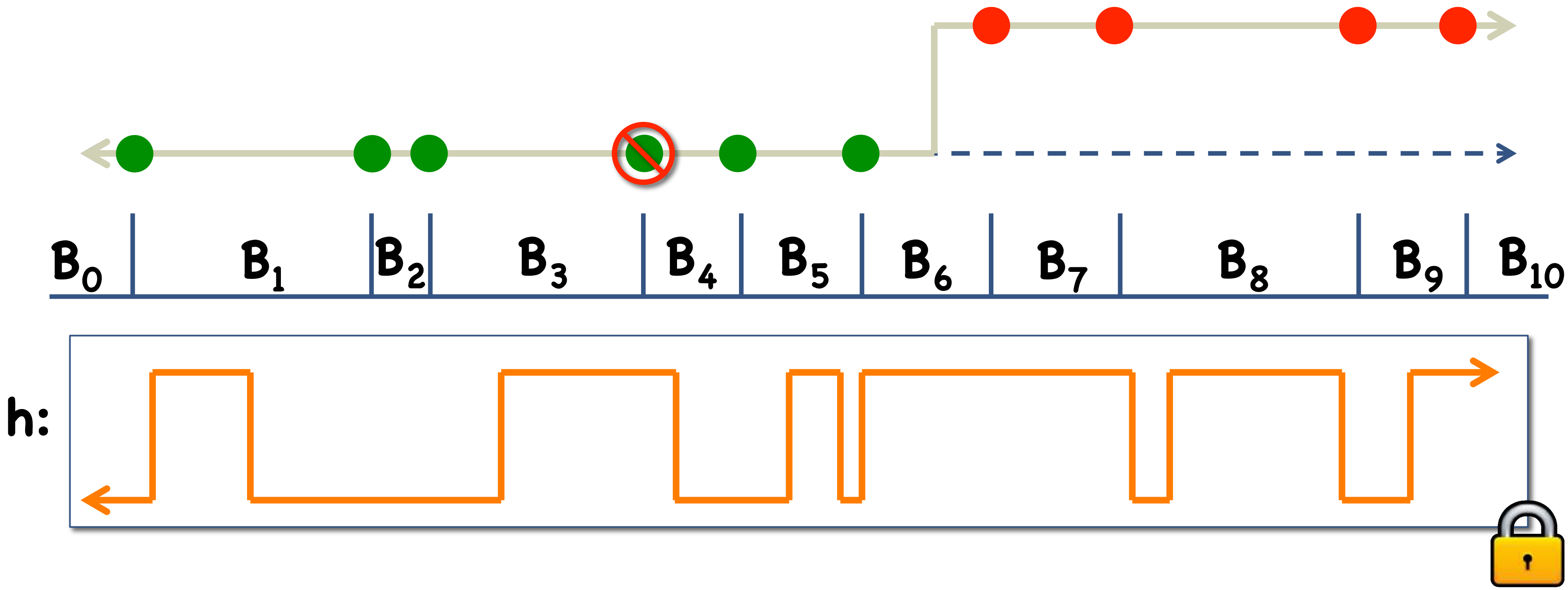
Goal: “Trace” learner, identify one of the samples



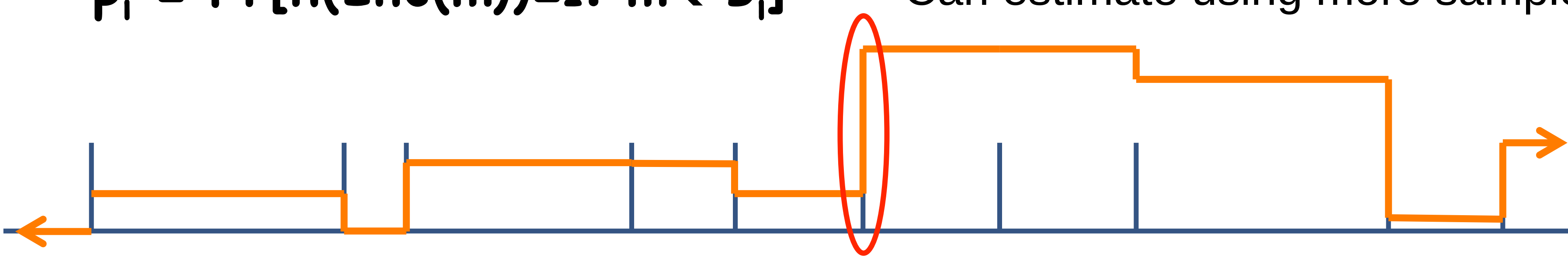
Tr is “good” (breaks differential privacy) if:

- Trace to some **c**
- Approx. correct **h** ⇒ **c≠c***

Re-identification for Encrypted Threshold



p_i = $\Pr[h(\mathrm{Enc}(m))=1:\ m \leftarrow B_i]$ ← Can estimate using more samples



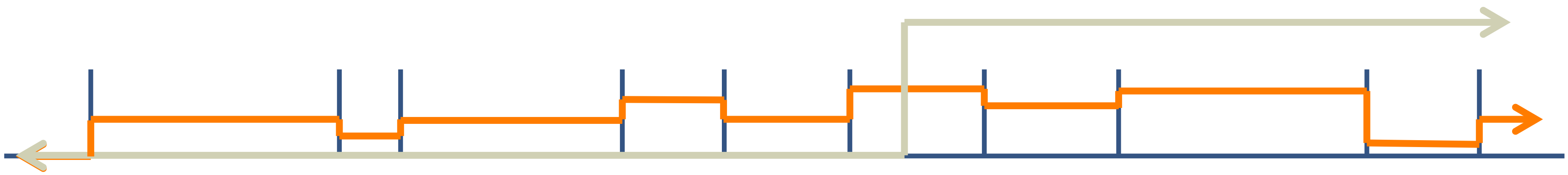
Output point with largest positive jump

Analysis

Tr always outputs some **c** ✓

Claim: **h** is approx. correct ⇒ some "large" positive gap

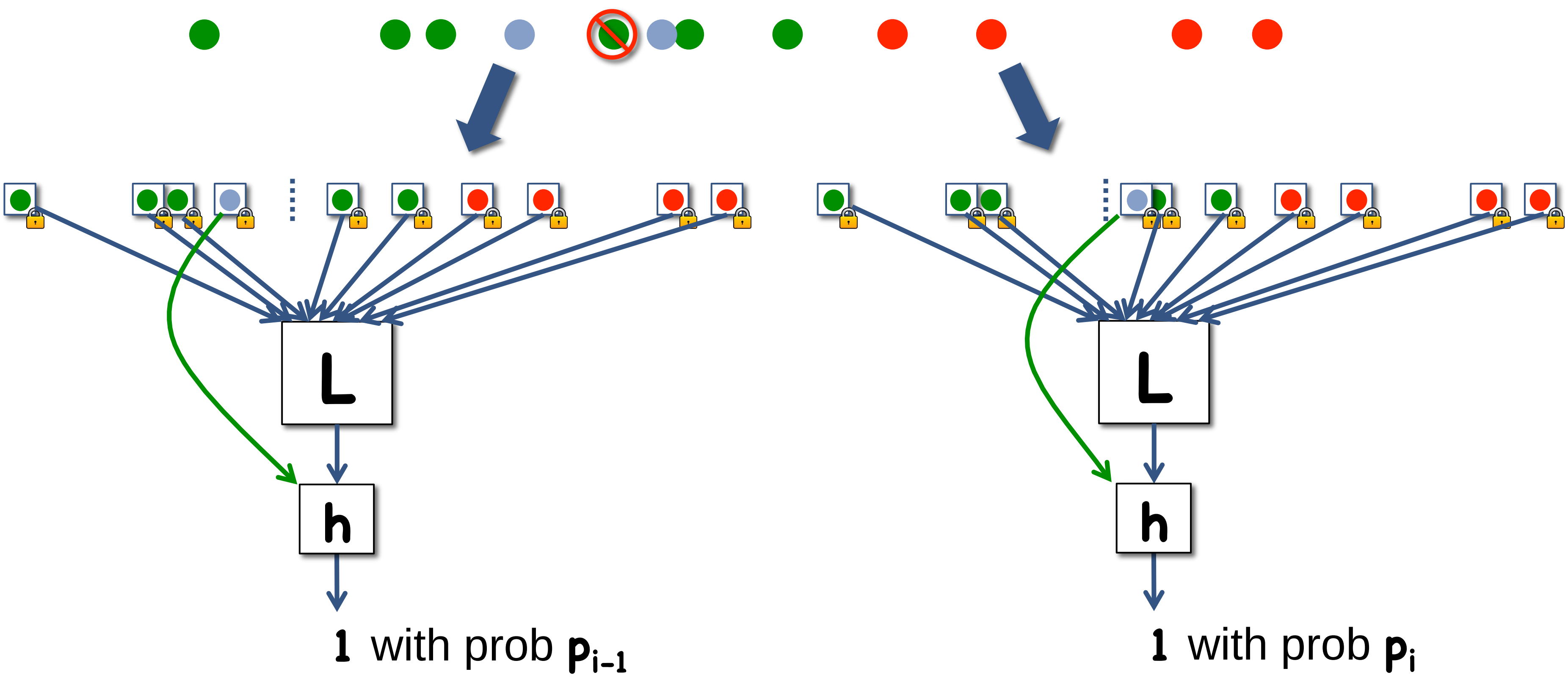
- No "large" positive gap ⇒ **h** poor approximation



Goal: show that large gap at **c*** breaks security

Call **h** "bad" if large gap at **c***

Analysis



"Bad" **h** ⇒ positive distinguishing advantage

- If **h** always "bad", overall positive advantage
- **Problem:** "good" **h** can have $p_{i-1} > p_i$ ⇒ overall advantage could be **0**
- **Solution**: different challenge set/analysis

Result

Thm: Assuming ORE (with strong correctness), there are efficiently PAC learnable concept classes that are not efficiently differentially privately learnable

How reasonable an assumption is ORE?

Constructions of ORE

In bounded **#(ctxt)** setting, can build from OWF:

- [GVW'12] bounded collusion FE from OWF
- [BS'15] Add function privacy
- **ORE.ctxt = FE.ctxt + FE.sk**

Unfortunately, we need unbounded **#(ctxt)**

- **#(samples)=#(ctxt)** should be independent of **C**
- For bounded **#(ctxt), C** depends on **#(ctxt)**

Constructions of Unbounded ORE

All known constructions use multilinear maps

- Through obfuscation [GGHRSW'13]
- Through FE [GGHZ'14] + [BS'15]
- Through multi-input FE [BLRSZZ'15]

Issue: All existing schemes have weak correctness

- Use current noisy maps [GGH'12]
- Come ciphertexts (those with large noise) cause comparison errors

Thm: ORE w/ weak correctness + Perfectly sound NIZKs ⇒ ORE w/ strong correctness

Constructions of Unbounded ORE

All known constructions use multilinear maps

- Through obfuscation [GGHRSW'13]
- Through FE [GGHZ'14] + [BS'15]
- Through multi-input FE [BLRSZZ'15]

Issue: Multilinear maps have unproven security

- [GGH'12,GGH'14]: "source group" assumptions broken
- [CLT'13]: Completely broken [CHRLS'15]
- [CLT'15]: Tweak to [CLT'13]. Is it really secure?

Constructions of Unbounded ORE

All known constructions use multilinear maps

- Through obfuscation [GGHRSW'13]
- Through FE [GGHZ'14] + [BS'15]
- Through multi-input FE [BLRSZZ'15]

Issue: Multilinear maps are very inefficient

- [BLRSZZ'15]: Best ORE construction
  - 16-bit plaintext → |ctxt| ≈ 23GB
  - 64-bit plaintext → |ctxt| ≈ 1.4TB

Using [ACLL'14] with $\lambda = 80$

Removing Mmaps from ORE

Conjecture: OWF insufficient for (unbounded) ORE

Conjecture: Bilinear maps insufficient for ORE

Conjecture: Constant arity mmaps insufficient for ORE

Hope: LWE sufficient for ORE?

Decreasing confidence

Thanks!