Hardness of Non-Interactive Differential Privacy from One-Way Functions

Lucas Kowalczyk
Tal Malkin

Jonathan Ullman
Daniel Wichs

Northeastern University

Results:

Hardness results for answering statistical queries with differential privacy

From Traitor-Tracing to Differential Privacy Hardness:

traitor-tracing scheme → differential privacy hardness result

[DworkNaorReingoldRothblumVadhan09]

iO + OWF → traitor-tracing scheme

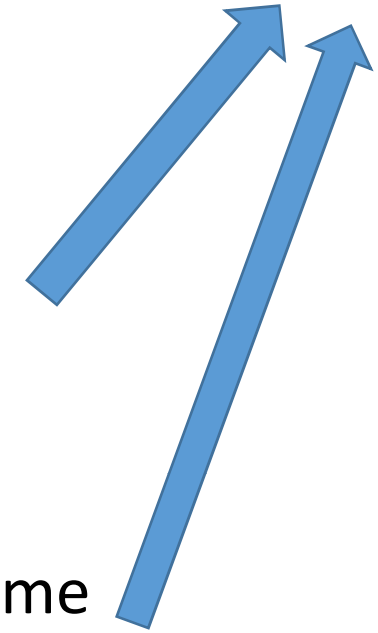
[BonehZhandry14]

Bilinear Groups → *risky* traitor-tracing scheme

[GoyalKoppulaRussellWaters18]

OWF → *even weaker* traitor-tracing scheme

[KowalczykMalkinUllmanWichs18]



Answering Statistical Queries with Differential Privacy:

Consider a dataset $D \in X^n$ where each of the n elements is some user's data, and each individual's data comes from some data universe X

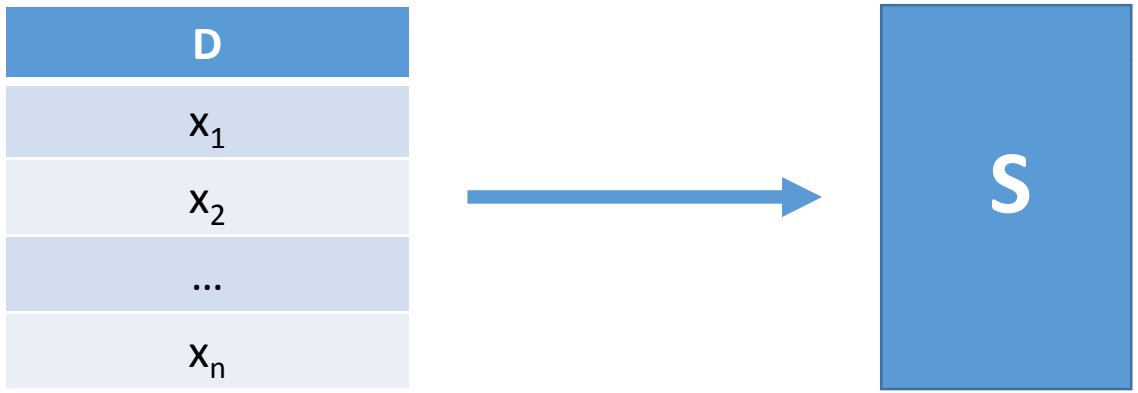
We'd like to be able to efficiently answer statistical queries on D, which are queries of the form:

"What fraction of individuals in D satisfy predicate p?"

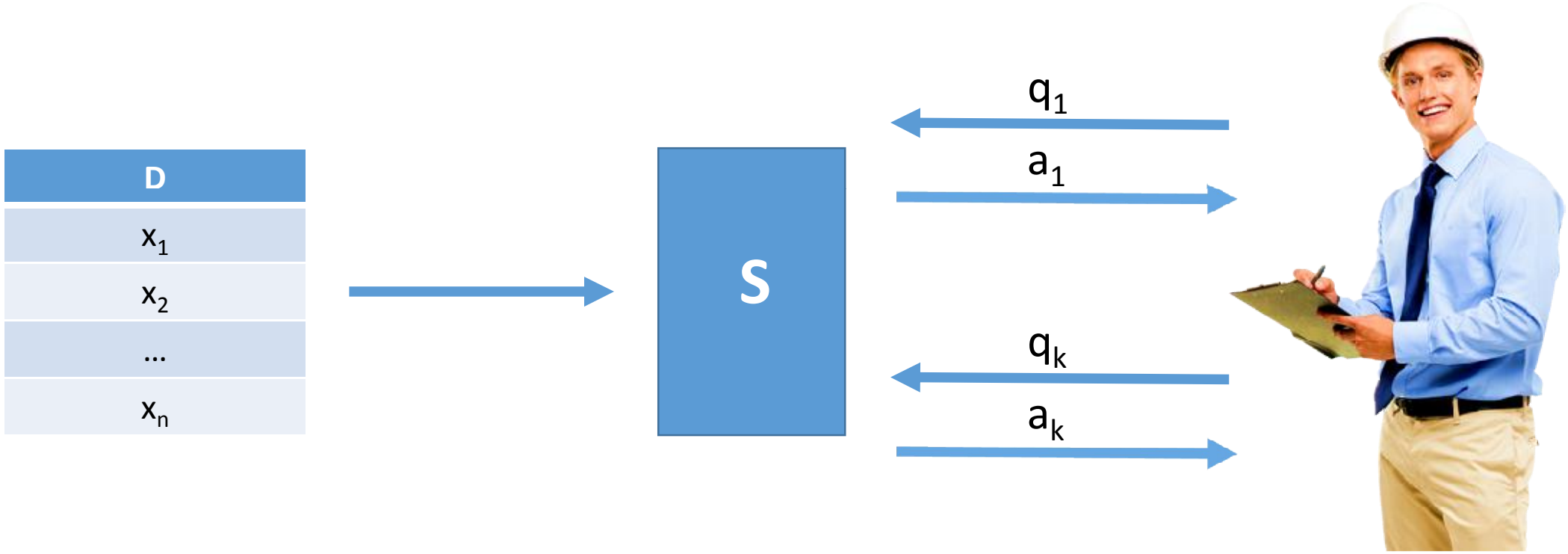
for p in some query set Q.

Differential privacy requires that we do so in such a way that no one individual's data has significant influence on the answers.

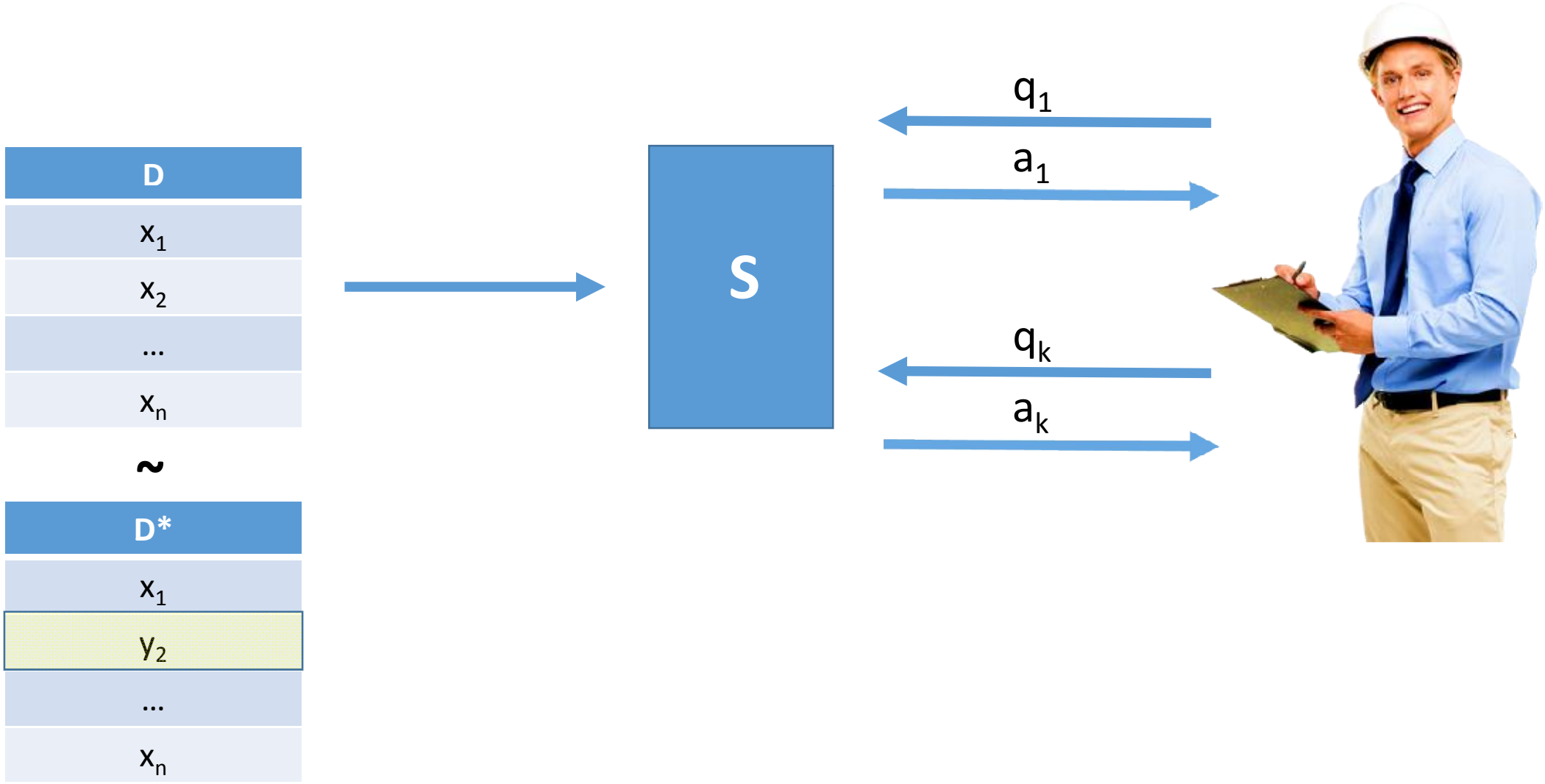
Differential privacy



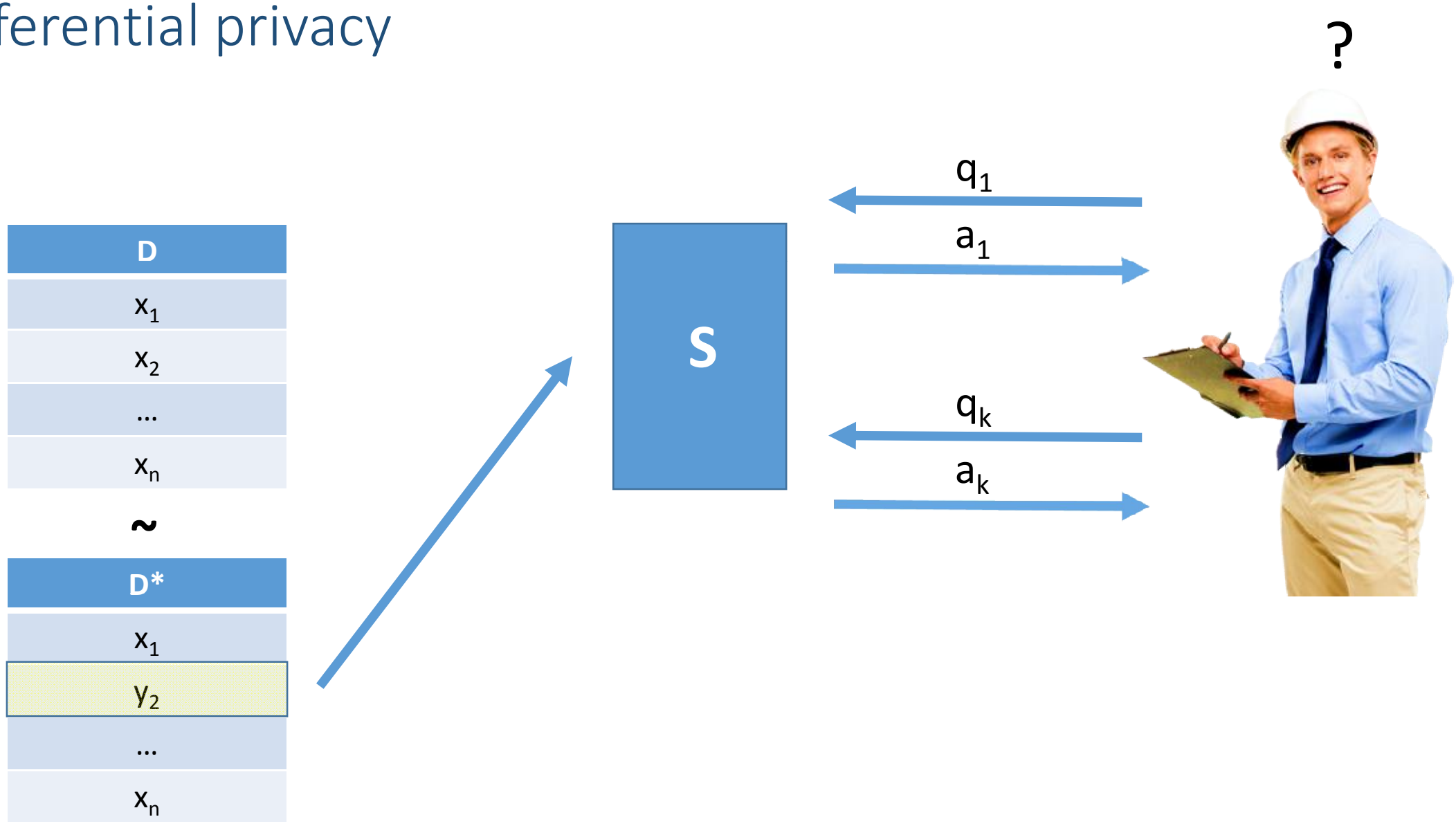
Differential privacy



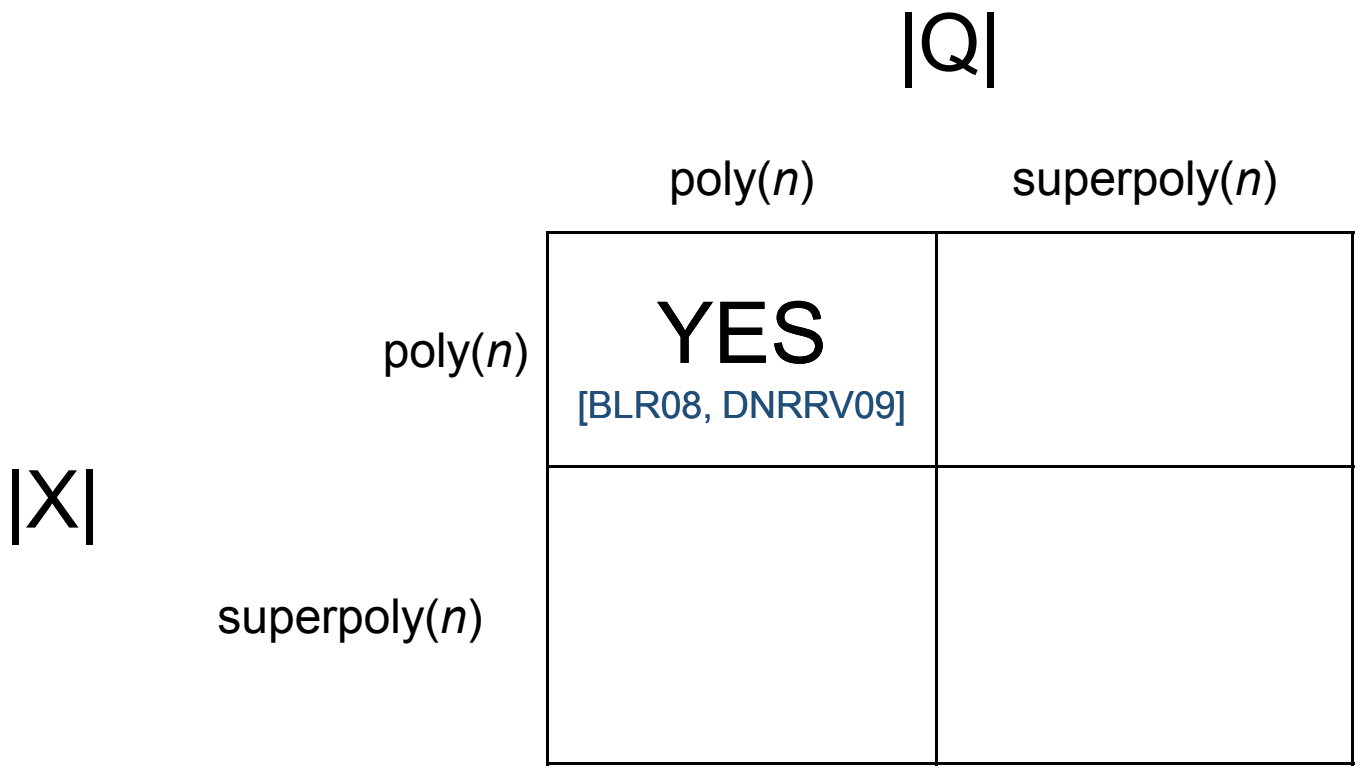
Differential privacy



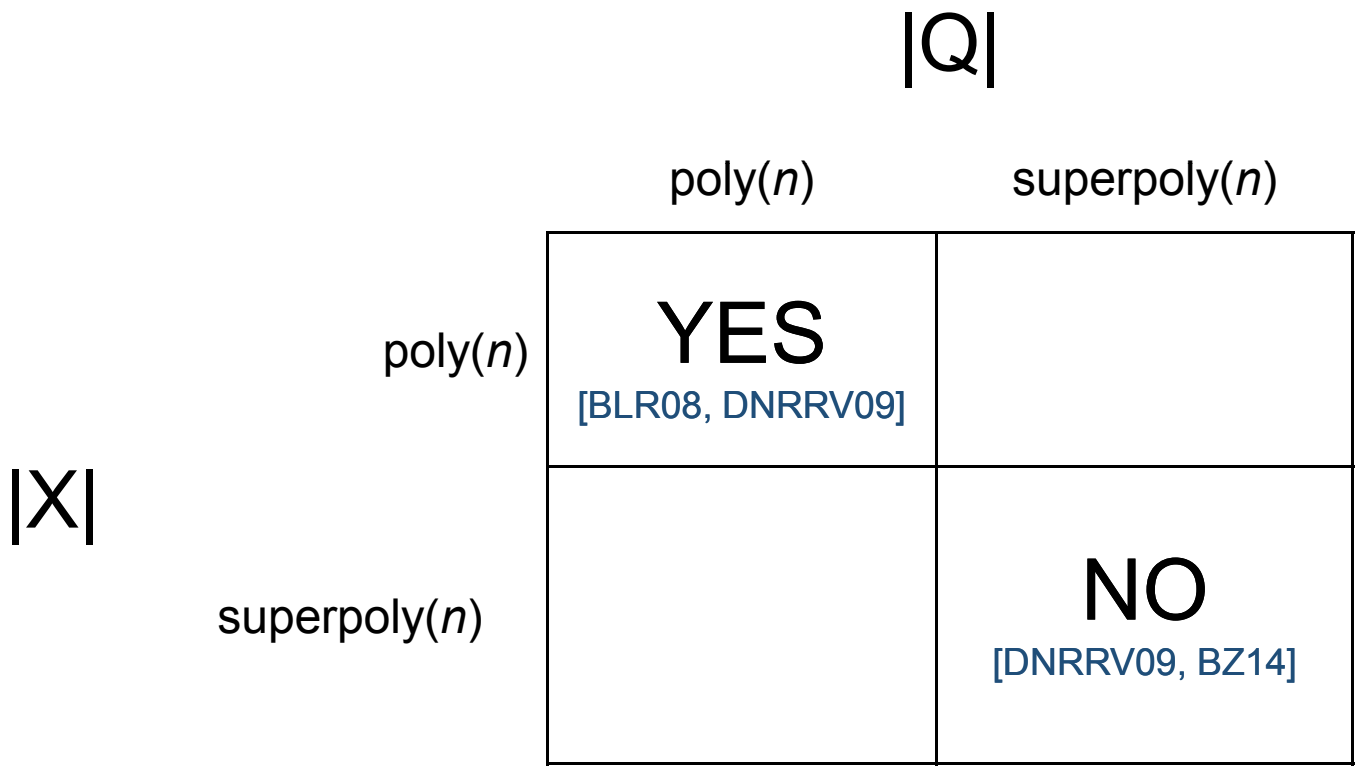
Differential privacy



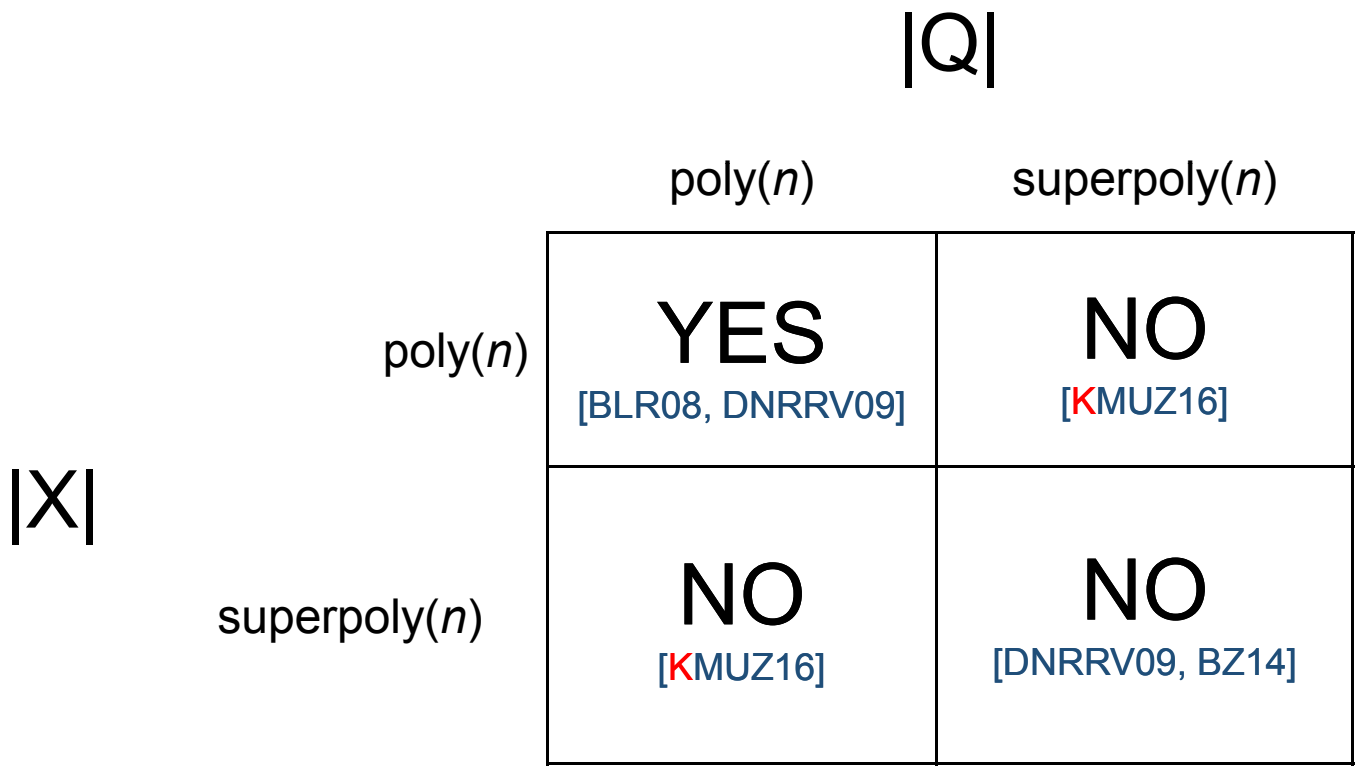
Can we efficiently answer statistical queries with diff. privacy?



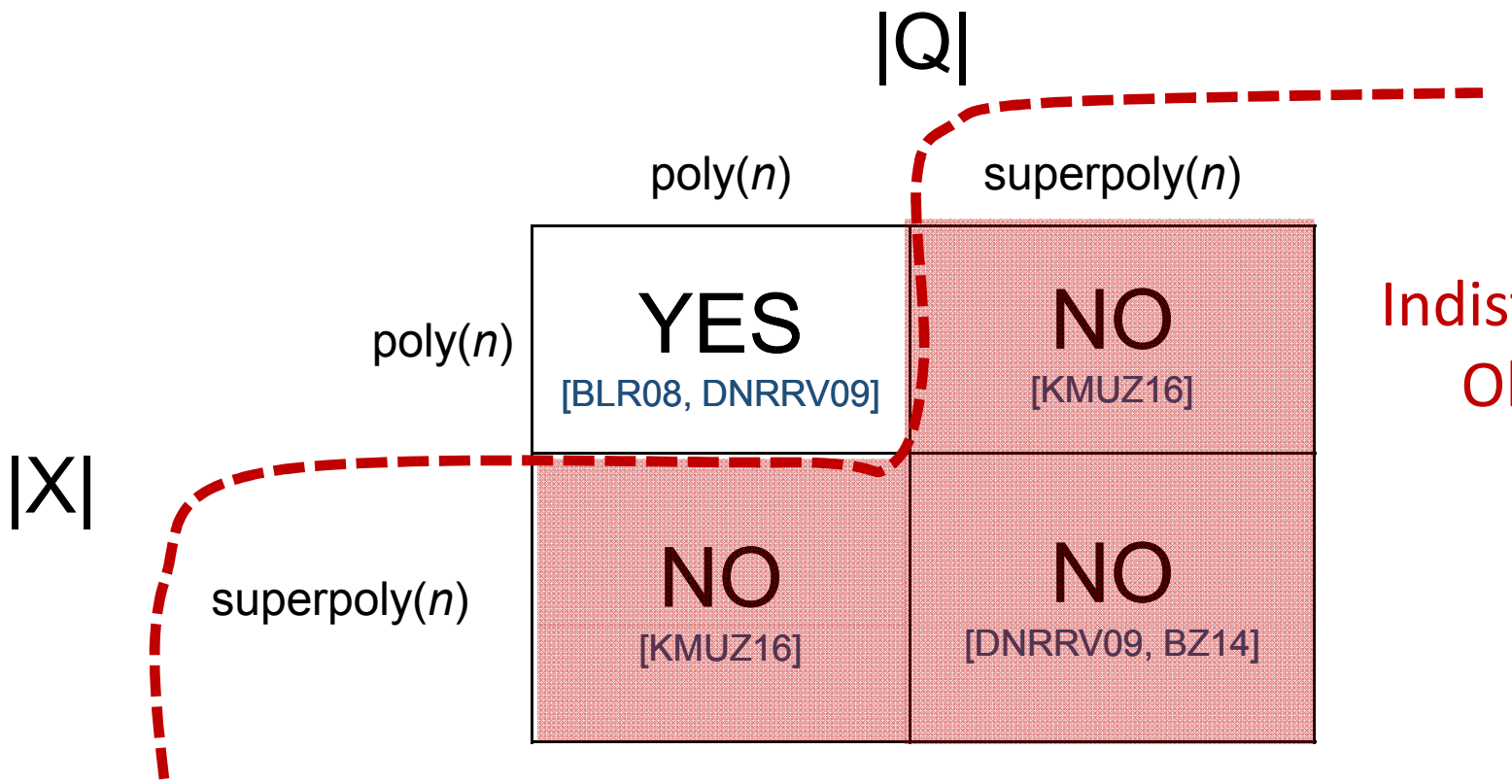
Can we efficiently answer statistical queries with diff. privacy?



Can we efficiently answer statistical queries with diff. privacy?

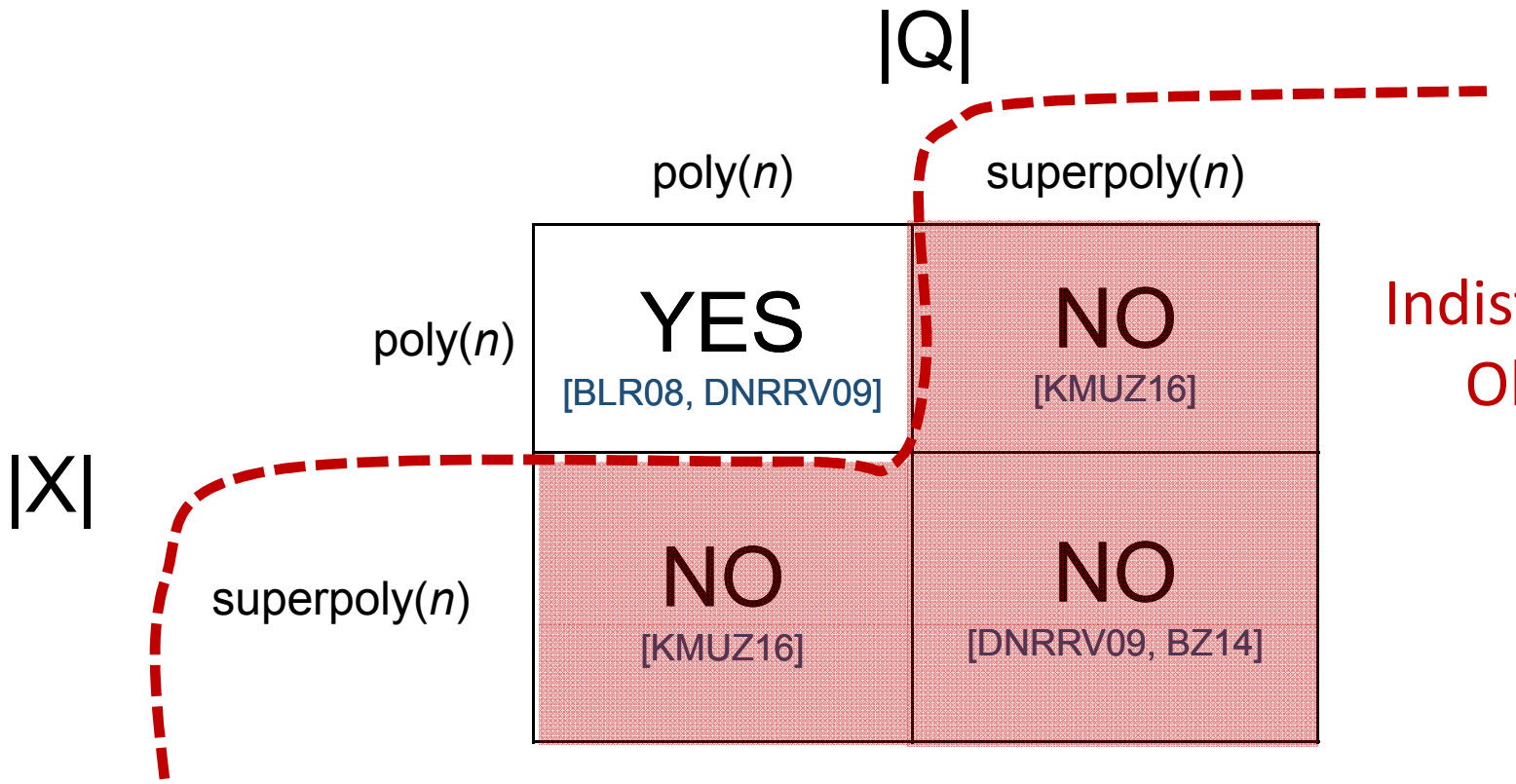


Can we efficiently answer statistical queries with diff. privacy?



Assume Indistinguishability Obfuscation!

Can we efficiently answer statistical queries with diff. privacy?

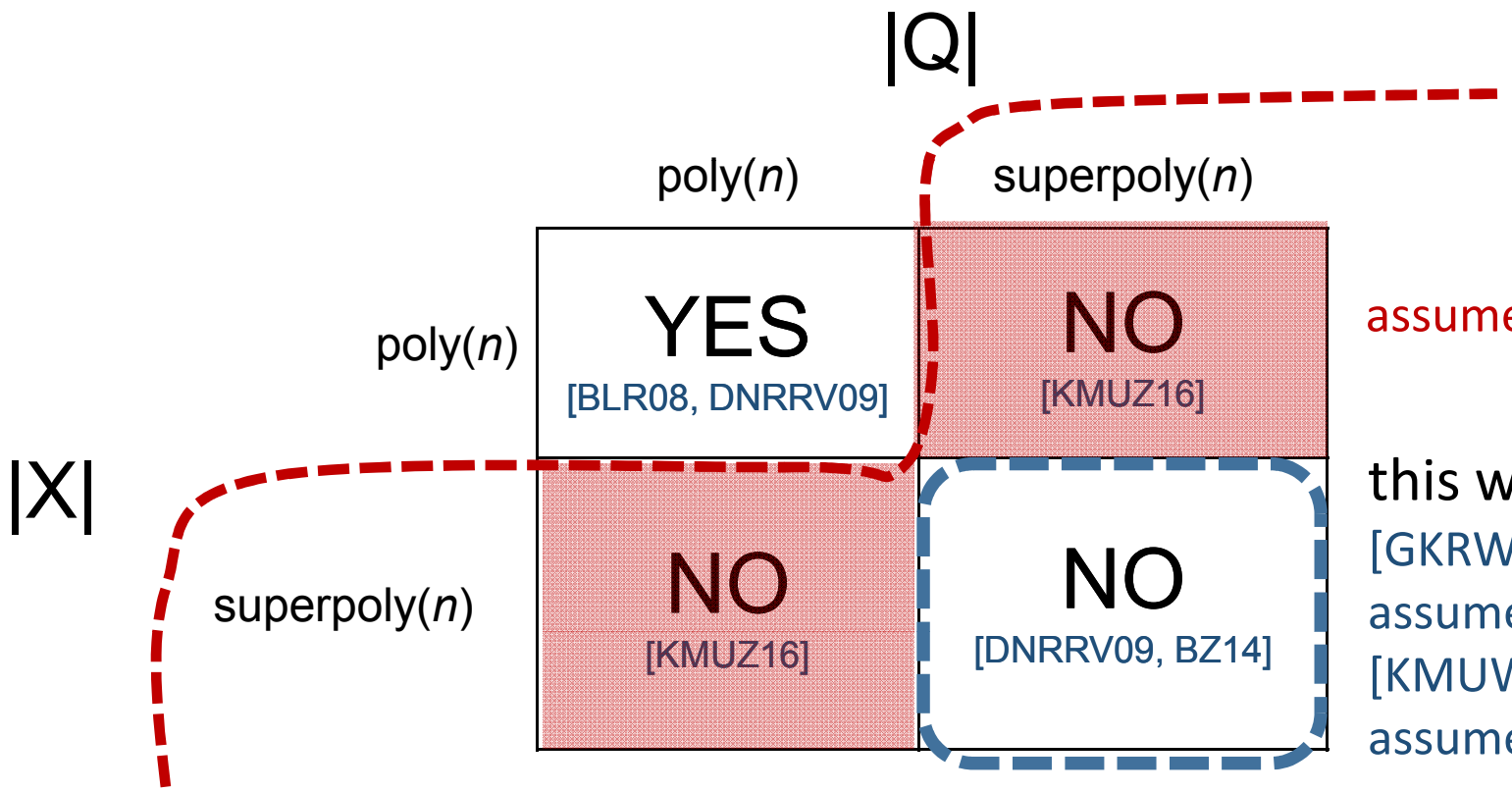


Assume Indistinguishability Obfuscation!

"Open Problem: Can a hardness result like [any of above] be established under a more standard and widely believed complexity assumption?"

– Salil Vadhan, 2016

Can we efficiently answer statistical queries with diff. privacy?



assumes iO + one-way functions

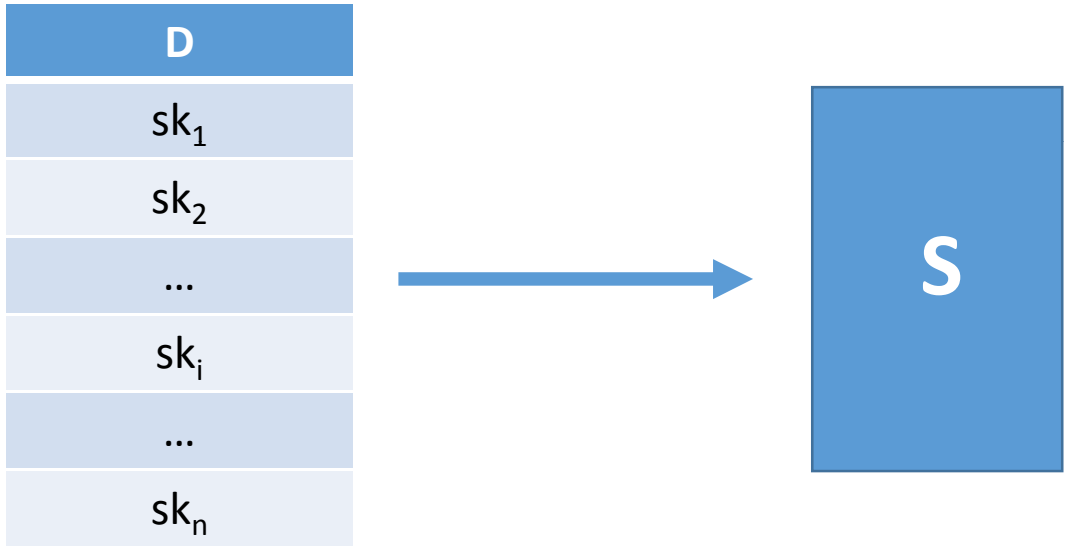
this work
[GKRW18]
assumes bilinear maps
[KMUW18]
assumes one-way functions

"Open Problem: Can a hardness result like [any of above] be established under a more standard and widely believed complexity assumption?"

– Salil Vadhan, 2016

Traitor-tracing Lower Bound [DNRRV09]

$|X| = 2^{|SK|}$

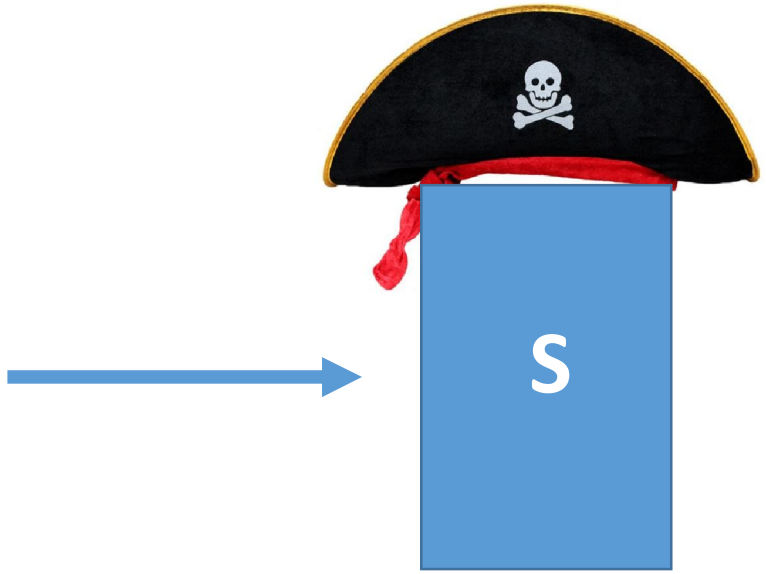


q_C = "what fraction of database decrypts ciphertext *C* to 1?"

$|Q| = 2^{|CT|}$

Traitor-tracing Lower Bound [DNRRV09]

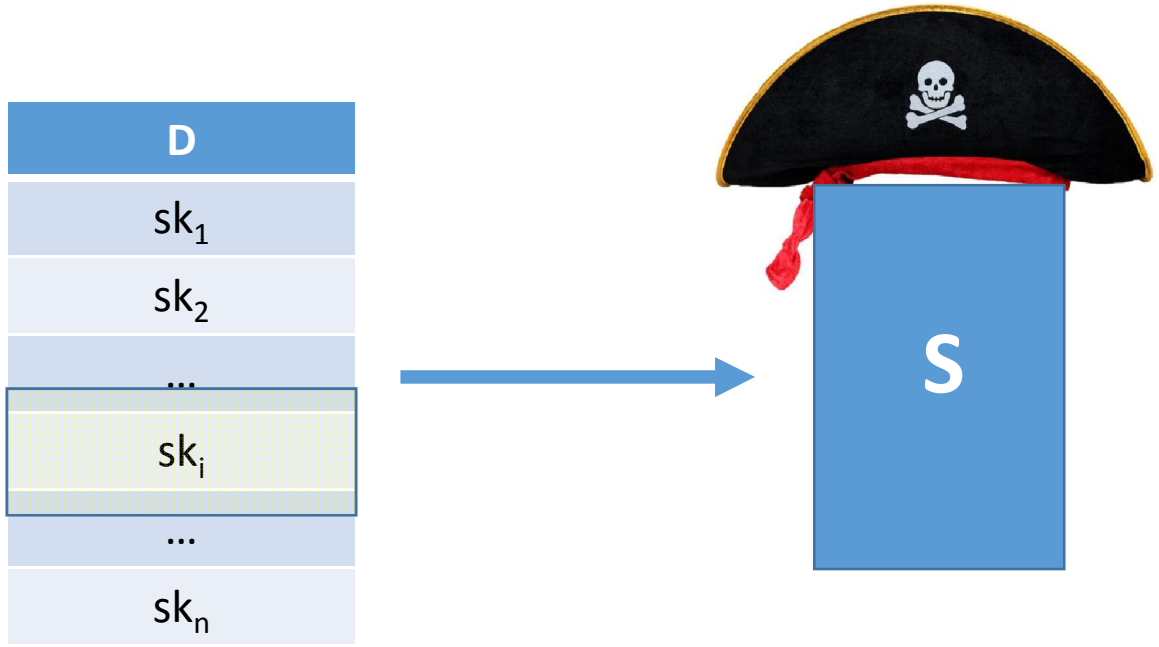
D
sk_1
sk_2
…
sk_i
…
sk_n



S is a pirate decoder!

q_C = “what fraction of database decrypts ciphertext *C* to 1?”

Traitor-tracing Lower Bound [DNRRV09]



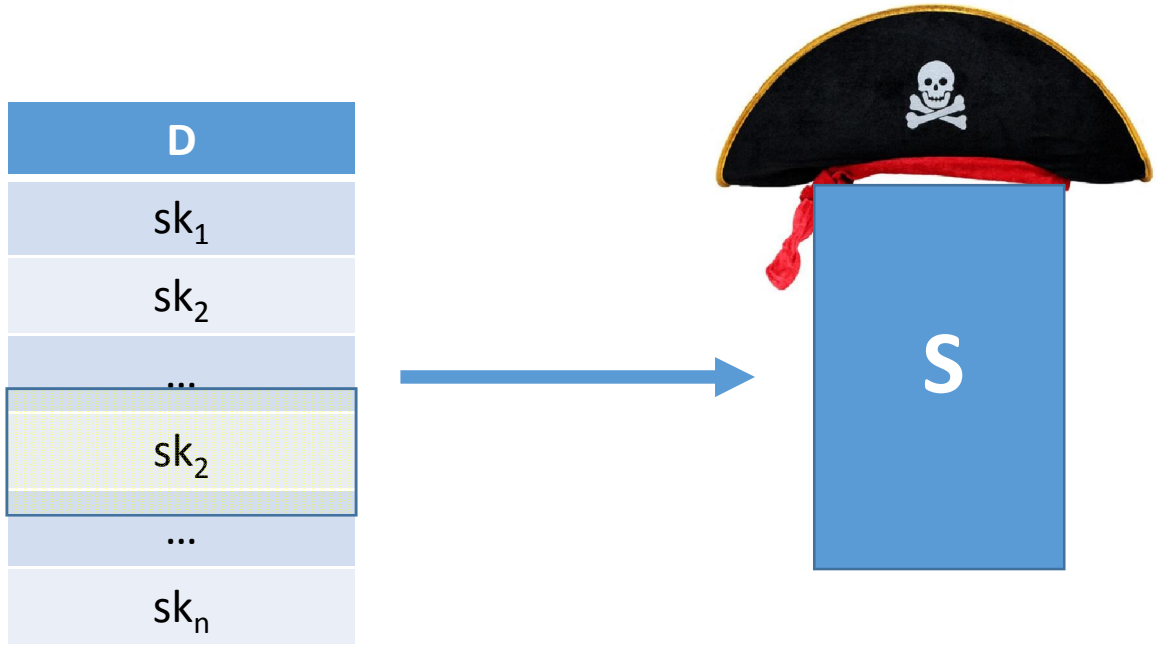
S is a pirate decoder!

Trace(**S**) = i

with probability $\geq \frac{1}{n}$

q_C = "what fraction of database decrypts ciphertext *C* to 1?"

Traitor-tracing Lower Bound [DNRRV09]



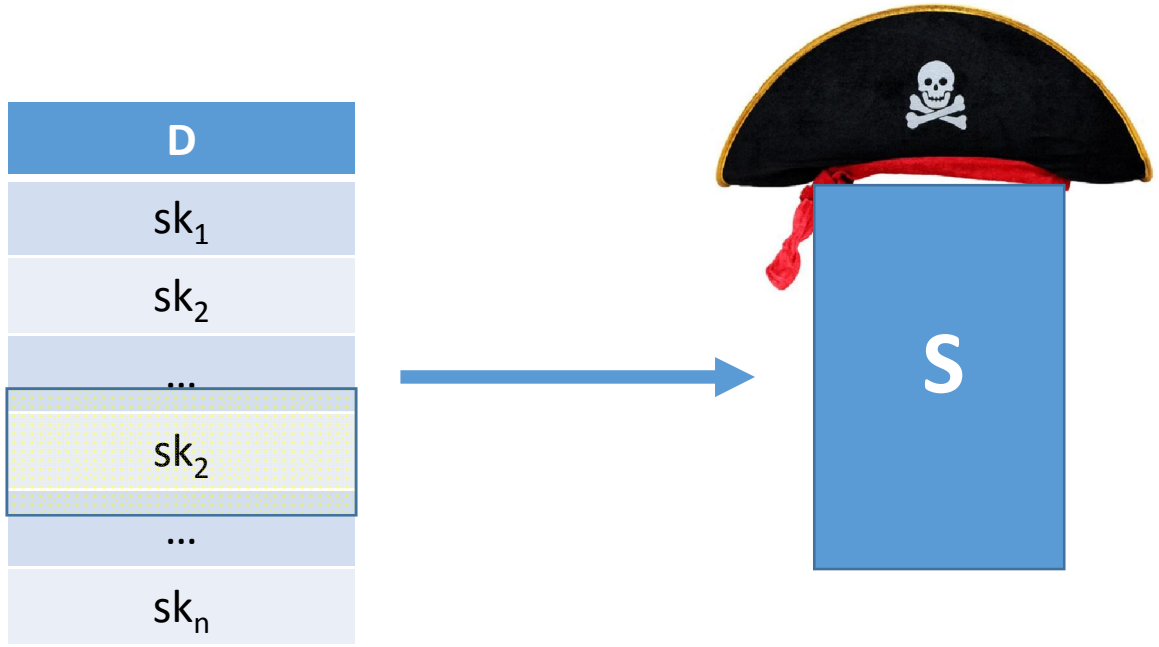
S is a pirate decoder!

Trace(**S**) = i

with probability $\geq \frac{1}{n}$

q_C = "what fraction of database decrypts ciphertext *C* to 1?"

Traitor-tracing Lower Bound: Take 2 [DNRRV09] / [GKRW18]



S is a pirate decoder!

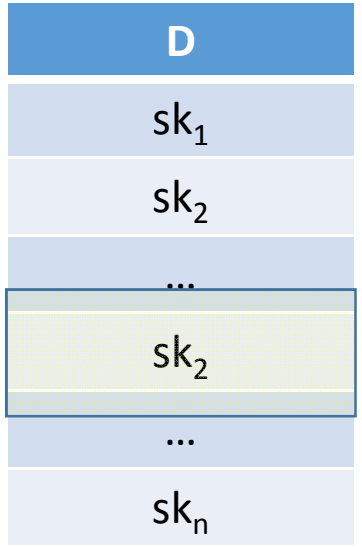
RiskyTrace(**S**) = i

with probability $\geq \frac{1}{n}$

q_C = "what fraction of database decrypts ciphertext *C* to 1?"

$$|X| = 2^{|sk|} = 2^{\lambda} \qquad |Q| = 2^{|ct|} = 2^{\lambda}$$

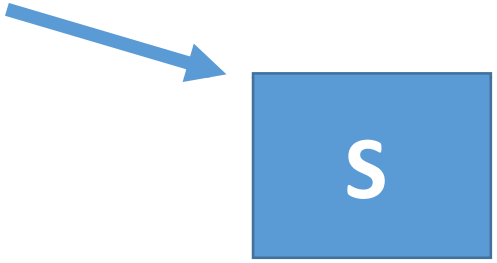
Traitor-tracing Lower Bound: Take 3 [DNRRV09] / [KMUW18]



-Traitor-tracing scheme need not be public-key!

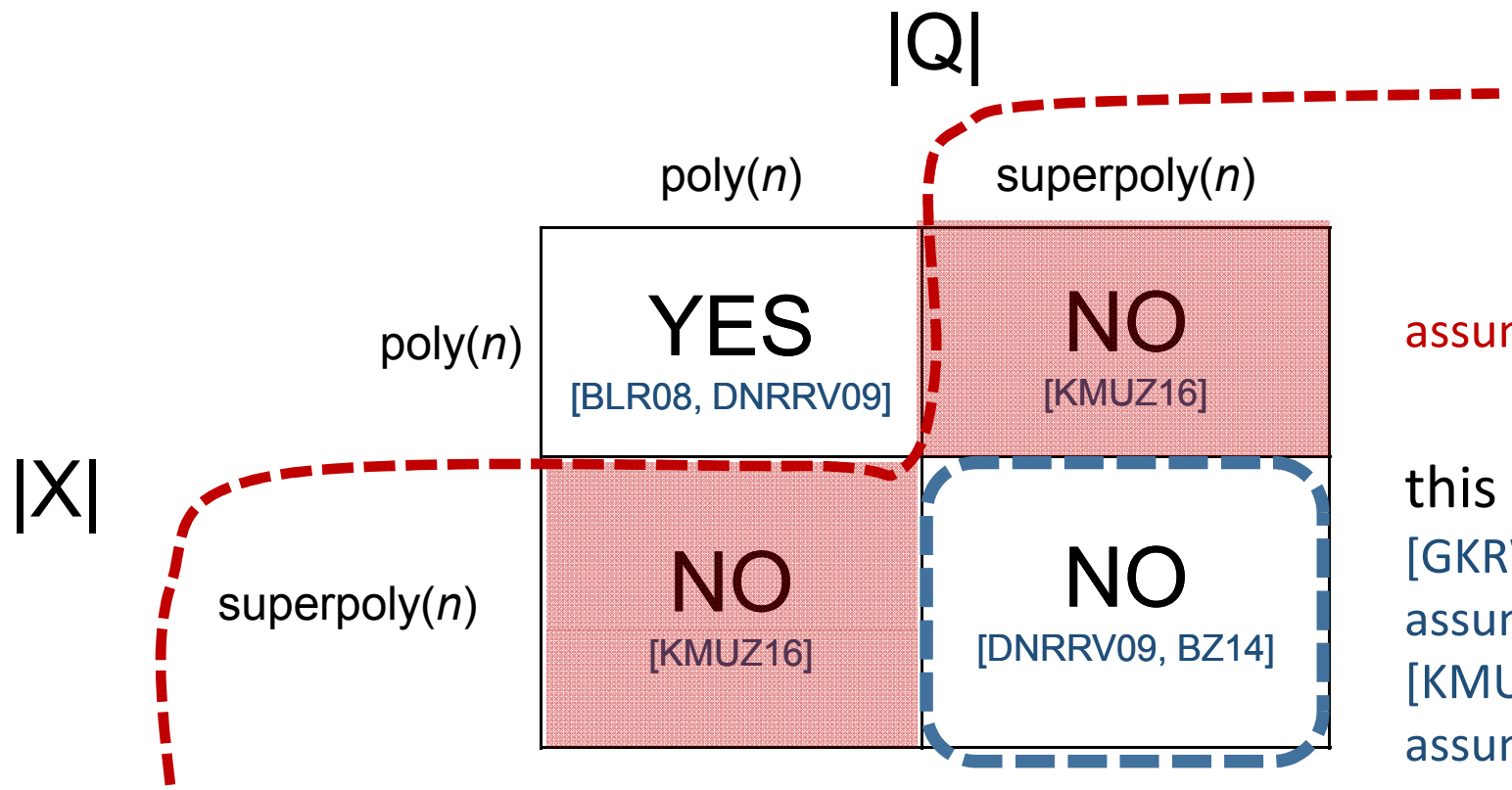
-**S** is created without any knowledge of ciphertexts

-achievable from functional encryption for comparisons via modified construction of [GVW12]



q_C = “what fraction of database decrypts ciphertext *C* to 1?”

Recap: Can we efficiently answer statistical queries with diff. privacy?



assuming iO + one-way functions

this work
[GKRW18]
assuming bilinear maps
[KMUW18]
assuming one-way functions

thank you!